DOCUMENT RESUME

ED 272 177 IR 012 246

TITLE	ADCIS Conference Proceedings (27th, New Orleans, Louisiana, February 3-6, 1986).
INSTITUTION	Association for the Development of Computer-based Instructional Systems.
PUB DATE	Feb 86
NOTE	311p.
PUB TYPE	Collected Works - Conference Proceedings (021)
EDRS PRICE	MF01/PC13 Plus Postage.
DESCRIPTORS	*Computer Assisted Instruction; *Computer Software; *Computer Uses in Education; Elementary Secondary Education; Higher Education; *Interactive Video; Media Research; Microcomputers; Program Descriptions; *Teaching Methods; Technological Advancement
IDENTIFIERS	Association for Devel Computer Based Instr Systs

ABSTRACT

The 52 papers in this volume, which represent recent research and applications in the field of computer-based instruction, are grouped under 10 general topic areas: (1) computer-based training; (2) elementary, secondary, junior college, and math education; (3) health; (4) home economics; (5) implementation; (6) mini-microcomputers; (7) PILOT; (8) PLATO users group; (9) interactive video-audio; and (10) theory and research. More than 60 200-word abstracts for additional topic presentations are included, as are lists of the Association for the Development of Computer-Based Instructional Systems (ADCIS) Board of Directors and Special Interest Group Council, and a proceedings index. (THC)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

U.S. DEPARTMENT OF EDUCATION
Office of Educational Research and Improvement
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)

☒ This document has been reproduced as received from the person or organization originating it.

☐ Minor changes have been made to improve reproduction quality.

- Points of view or opinions stated in this document do not necessarily represent official OERI position or policy.

ED272177

ADCIS

IR012246

"PERMISSION TO REPRODUCE THIS MATERIAL HAS BEEN GRANTED BY
J.J. Hirshbuhl

TO THE EDUCATIONAL RESOURCES

27th ADCIS CONFERENCE PROCEEDINGS

Association for the Development of Computer-Based Instructional Systems

NEW ORLEANS, LOUISIANA
FEBRUARY 3 thru 6, 1986

Copyright ADCIS 1986
ADCIS, MILLER HALL 409
WESTERN WASHINGTON UNIVERSITY
BELLINGHAM, WASHINGTON 98225

FORWARD FOR PROCEEDINGS

ADCIS is proud to present these papers representing the most recent research and applications in the field of computer-based instruction. It is our intent that this document contribute to the advancement of the education and training professions. The works presented herein are the result of many dedicated hours colleagues have given so that you may benefit. As you use this document and meet the many writers, acknowledge their work and let them know it is appreciated. It is their leadership that enables us to advance quickly in improving the quality and efficiency of learning for everyone.

Glenn E. Head
Program Chairperson

ADCIS BOARD OF DIRECTORS

President
Ron Comer
The Ohio State University

Program Chairperson
Glenn Head
Instructional Communications

JCBI Editor
Allen Avner
University of Illinois

Newsletter Editor
Tom Reeves
Instructional Technology

Special Interest Group Chairperson
Bob Tannenbaum
Hunter College

Special Interest Group Representative
Terris Wolff
University of Southern California

ADCIS Executive Director
Gordon Hayes
ADCIS International Headquarters
Ex-officio

SPECIAL INTEREST GROUP COUNCIL

Computer-Based Training
Robert Fratini
AT&T Technologies

Elementary/Secondary/Jr. College/Math
Chris Dimas
Kennedy King College

Educators for the Handicapped
Susan Rice
University of Minnesota

Health Education
Arthur D. King
University of Florida

Home Economics
Nancy Dillon
Phoenix, Arizona

Implementation
Terris Wolff
University of Southern California

Interactive Instructional System Presentation and Authoring
Sue Leslie
IBM Corporation

Interactive Video/Audio
Donna Larson
Grand Valley State Colleges

Mini/Micro
Bob Tannenbaum
Hunter College

Music
Michael Arenson
University of Delaware

PILOT
W. Kirk Richardson
Georgia State University

PLATO
Peter Johnston
University of Alberta

Theory & Research
Frank Roberts
Control Data Corporation

TABLE OF CONTENTS

COMPUTER BASED TRAINING (SIG/CBT)

A Blueprint for CBT Courseware Development 1
Gregory C. Hamilton
Making Feedback in Computer-Based Training Meaningful 5
John C. LeDoux, Charles J. Stanley
Advertising Sponsorship of Software: Who Could Benefit? 9
James MacLachlan
Microcomputers in Business Program 13
Jane A. Malloy
State Notation and a Top-Down Instructional Design Methodology 17
Kenneth Stephens

ELEMENTARY, SECONDARY, JUNIOR COLLEGE, MATH EDUCATORS (ELSECJC/MATH)

An Investigation of the Staff Concerns in a School District Implementing Computer Education 22
Dennis D. Cole
Reducing Learning Anxieties With Computer-Based Instruction 27
Albert L. Harris, Jacqueline M. Harris
1000 Requests/Week on Limited Resources 33
Joan Hayes
A Plan for Computer Literacy 36
Jan Powell
Pilot Project: LOGO for Preschoolers as an Aid for Teacher Preparation in Technology 39
Charlotte Scherer

HEALTH (HESIG)

The Uses of CBE in Teaching the Functions of Complex Systems 43
Joel A. Michael, Allen A. Rovick
Substance Abuse CAI: A Computer-Based Approach to Primary Prevention 49
Barbara S. Thomas, Greg Saunders, Ron Roskam

HOME ECONOMICS CONSORTIUM (SIG/HOMEC)

Attitudes Towards a Computer Simulation to Assess Quality of Ready-to-Wear 52
Ruth Marshall
A Proposed Model for Field Testing 53
Frances Whitaker Mayhew
Future Growth and Development of Microcomputers in Home Economics 58
Louise Wesswick

IMPLEMENTATION (ISIG)

CAI as Part of the Total Faculty Development Package 64
Kathleen Bellinger
The Use of a Graphic Scratchpad for Students in ICAI 68
Dick J. Bierman, Anjo A. Anjewierden
Factors of Flexible CMI System Design 72
Robert L. Jorczak
A Microcomputer-Based CAI Teaching Materials Production Support System 78
Masahiko Kurata, Hiroyasu Chimura, Yasuhiro Morimoto, Takahiro Sato
TAIGA, A Graphic Authoring System 84
Eelco F. Laagland
The Plan: An Instructional Systems Design Approach to the Analysis and Design of Technical Documentation 90
Patricia Anne Lawson
A Model for Supporting University CBI Products and Services 96
P. L. Hyde, D. G. Mellor, C. A. Parker, C. B. Phillips, J. H. Wilson

TABLE OF CONTENTS *(cont.)*

User-Oriented Tools for CBI Development 101
George A. Reed
Needs Analysis for a Computer-Based Statistics Course 105
Mary Jac Reed
Theory VS Reality: Application and Modification of Instructional Design to Training Situations 108
Richard Reinhart
The Implementation of a CBT Course for a Software Package that Monitors the Performance of an Operating System 110
Janis M. Stewart, Emile E. Attala
Intelligent Tutoring Systems: An Explanation 115
Kathleen M. Swigger, Steven F. Brown

MINI-MICRO

Operation of a Microcomputer Classroom Using a Network Application 120
Nancy Balogh, Carol Jarom
Several Strategies to Maximize the Effectiveness of Computer Assisted Instruction 123
George H. Conklin
Developing CAI in Pascal 127
Louisa Frank
Tools for Developing a Geography Package in UCSD p-System 131
Carol A. Jarom, Paige Vinal
Computer User Groups: An Untapped Resource 136
LaVerne L. Ludden
Developing a Library Skills Program for the IBM PC 144
Deborah E. Richards
Some Considerations Involved in the Installation of Ethernet Networks at the University of Delaware 148
George A. Reed, Nancy Balogh
Writing and Producing Effective Microcomputer User Guides 151
Chris Scherer

PILOT

A Comparison Study of Math Attitudes in Programming Classes of PILOT, Logo and BASIC 155
Barbara Signer

PLATO USERS GROUP (PUG)

Problem Solving Oriented CAI in University Statics Courses 158
Helen Kuznetsov

INTERACTIVE VIDEO-AUDIO (SIG/IVA)

A Videotex in Your Future? 164
Elisabeth M. Craig
Simple Interactive Video Disc Experiments of Motion in Physics 168
Ronald Gautreau
Evaluating Interactive Videodisc Instruction: A Procedural Model 171
Janis Morariu
Designing Blueprint Videodiscs for Flexible, Multilevel and Future Use 175
Sharyn Neuwirth, Janis Morariu

THEORY AND RESEARCH (SIG/TAR)

The Context of a CAI Lesson 180
J. D. Bradley, L. Glasser, S. W. Orchard
Teaching Cognitive Strategies to Native Students through Integrating Instrumental Enrichment and LOGO Programming 184
W. Bruce Clark, Kathleen Wenger

TABLE OF CONTENTS *(cont.)*

The Effects of Orienting Strategies and Cognitive Processing Time on Factual and Inferential Learning 190
M. J. Hannafin, T. L. Phillips, L. P. Rieber, C. Garhart

The Impact of Computer Technology on Programmed Instruction Materials in Speech Communication 195
Michael R. Hemphill, Christina C. Standerfer

The Courseware Instructional Toolkit: A Prototype Automated ISD System for Personal Computers 199
Greg Kearsley

Cognitive Style and the Interpretation of Computer Graphics 205
John David Knechtel

An Empirical Comparison of Menu Choice Mechanisms for Educational Computing Systems 210
Elaine Lewis

SMark: An Automated System for Grading Short Answers in Natural Language 214
Martin Janta-Polczynski, Gerald K. Rambally, Arjinder S. Uppal

Better Chemistry Through ICAI — Getting Started 221
Lynn H. Smith, Roland Garton

Learning Styles and Embedded Training: A Case Study 227
Mary L. Stoddard

A Model for Research, Design, and Selection of Istructional Uses of Technology in Mathematics and Science Education 234
Arthur L. White, Donna F. Berlin

TOPIC PRESENTATIONS . . . 200 Word Abstract (no formal papers)

Seven Guidelines for CAL ProjectTime Management 239
Raymond Ng

The Joint Staff Officer Training System (JSOTS) 240
Raye Newmen

An Instruction System Using Data Base Management Software 241
James D. Jensen

Approaches to Computer Simulations for Management Training 242
Stephen Greenfield, Gary Moehnke

Using Data Base Management System Software on Personal Computers for Instruction 243
Louis E. Frenzel, Jr.

How Artificial Intelligence Techniques Can be Used in Developing Courseware 244
Brian L. Dear

The Project to Evaluate Authoring Systems: A Progress Report 245
Victoria L. Clegg, Jane M. Lommel

Using Computers in Couselor Training 246
William E. Schulz

Computer Based Preventive Maintenance Training for the Bus Industry 247
Allan Williams

The Evolution of a Computer Lab: Tool of Technology Transfer in International Education 248
Claire Bradin, Larry Cross

Implementating Micro Computer Based CMI/MIS in K-12 Schools 249
Michael Burger

Computer-Based Language Experience Strategies for Kindergarten and First Grade Students 250
Claudia W. Cawthon, Thomas C. Reeves

The Faculty Retraining Program in Computer Science at Stevens Institute of Technology 251
Lawrence E. Levine

On Enjoying Teaching Algebra: Letting Teachers Do What They Do Best 252
John C. Miller

Teacher-"Whiz-Kid" Pairing: A Cooperative Model for Tech Ed Teacher Training 253
Lisa Novemsky

The Impact of Computer-Based Composition Instruction Activity on the Freshman English Curriculum 254
Raj Singha

The Computer: Technological Teammate for Teachers 255
Kay Walla

Improving Nurses' Technical Writing Skills with Computer Assisted Instruction 256
Robert Berner

TABLE OF CONTENTS *(cont.)*

Computer Skills for Future Elementary Teachers 257
Lynda Thomas, Linda Thompson
Pro-Nurse: A Software Program for Automating Individual Testing 258
Jean M. Arnold, William Greenhalgh
Simultaneous Computer Literacy and CAI Authoring 259
Lou Ann Emerson, Elizabeth Weiner
Academic Software - Vision and Perspective 260
Steven W. Gilbert
A Generic Tutorial Program Incorporating Videodisc Images and Text Paging 261
Ronald P. Jensh, F. Scott Beandenkopf
A Microcomputer Program of Pulmonary and Tissue Gas Exchange 262
Joseph Boyle III
Simulations in Respiratory Physiology Suitable for Use in Multiple Teaching Settings 263
H. I. Modell
Teaching a Computer Literacy Course in a Baccalaureate Nursing Program 264
Patricia J. Morin
Three Models for Designing Patient Care Simulations for Interactive Videodisc 265
Jo Ann Reynolds
Evaluation of a CAI Microcomputer Laboratory 266
Elizabeth Weiner, Lou Ann Emerson
Converting Instructional Videotapes to CAIV Format 267
Dent M. Rhodes, Janet White Asbell
Program Path Utilization of Nursing Students Using a Computer Assisted Tutorial 268
Marianne E. Yoder
Consistency vs. Monotony in Instructional Software 269
Marsha Young
Getting Involved with Computers with Limited Funds and Without a Computer 270
Nadine Hackler
Total Integration Moving Computers into the Department 271
Ann B. Kapp
Computer-Based Models for Dietetics Education 272
Lois Schroder, Phyllis Kent
Course Production Problems on Mini and Micro Computers 273
Jeffrey Howell
Looking Up: A Funder's Perspective on the Post-Secondary Uses of Technology 274
Stephen C. Ehrmann
Integrating Personal Computers into a Supervisory Management Curriculum 275
LaVerne Ludden, Michael Miller
The Future of CAI Development by University Educators 276
Steve Hirst, Pete Trotter
Instructional Practice and Computer-Based Instruction an Implementation Model 277
Marvin O. Heller, John F. Wedman
Meeting Faculty Access to Instructional Computing Needs Utilization Trends for the University of Florida's Faculty Support Center for Computing 278
Wayne A. Wolfe
Developing Ear Training Skills Using a Microcomputer 279
Richard Shirey
A Computer-Aided Design Engineering Center 280
Donald W. Thomas
The Student Information Resource System and Interactive Videotape Project 281
Eric Halsey
The Interactive Video Institute: A Hands-on Approach to Teaching Interactive Video 282
David Burke
Integrating Orienting Strategies in the Design of Interactive Video 283
Michael J. Hannafin, Curtis Hughes
A Multi-level Guided Practice Tutorial Using Interactive Video 284
John F. Moore

TABLE OF CONTENTS *(cont.)*

Focus on L.O.C.O.S. Interactive Videodisk Training for Library Users 285
Thomas C. Reeves, William D. Aggen

An Overview and Plan for the Canadian Implementation of Computer Assisted Learning Courseware for Special Needs Children, Adolescents and Adults 286
Lauran Sandals

A Neurocognitive Approach to Teaching Reading: A Computer Assisted Program 287
Phyllis Anne Teeter, Philip Smith, Randall Ryder

CAI Guidelines for Dialogue Interaction 288
Aaron B. Navarro

Developing a CAI Tutorial with PC Pilot 289
Lynda Doll, W. Kirk Richardson, Henry W. Sottnek

Screen Design Techniques Using the Pilot Programming Language 290
Karen A. Droms

Teaching Pilot in CAI Academic Courses and Workshops 291
W. Kirk Richardson

The Effects of Orienting Strategies and Practice on the Learning of Cued and Uncued Information 292
Michael Hannafin

Authoring Software and Effective Computer Assisted Instruction 293
Cindy Leshin, Joellyn C. Pollock

The Accuracy of Students' Cognitive Monitoring During Computer-Based Instruction 294
Casey Garhart

Enhancing Students Creative Abilities Through the Use of the Microcomputer 295
J. Mark Tisone

The Fipse Technology Study Group and the Education Technologies Database 296
Mary Grace Smith

An Hierarchial Menu-Driven Microcomputer Network 297
G. Brink, L. Glasser

Pro-Nurse: A Software Program for Automating Individualized Testing 298
Jean M. Arnold, William Greenhalgh

Evaluation of a CAI Microcomputer Laboratory 299
Elizabeth Weiner, Lou Ann Emerson

PROCEEDINGS INDEX

A **Page**

Aggen William D. 285
Anjewierden, Anjo A. 68
Arnold, Jean .... 258 - 298
Attala, Emile E. 110
Azbell, Janet 267

B

Balogh, Nancy .... 120 - 148
Beadenkopf, F. Scott 261
Bellinger, Kathleen 64
Berlin, Donna, F. 234
Berner, Robert 256
Bierman, Dick J. 68
Boyle, Joseph III 262
Bradin, Claire 248
Bradley, J.D. 180
Brink, G. 297
Brown, Steven F. 115
Burger, Michael 249
Burke, David 282

C

Cawthon, Claudia W. 250
Chimura, Hiroyasu 78
Clark, W. Bruce 184
Clegg, Victoria L. 245
Cole, Dennis D. 22
Conklin, George H. 123
Craig, Elizabeth M. 164
Cross, Larry 248

D

Dear, Brian L. 244
Doll, Lynda 289
Droms, Karen A. 290

E

Ehrmann, Stephen C. 274
Emerson, Lou Ann .... 259 - 266 - 299

F

Frank, Louisa 127
Frenzel, Louis E. 243

G

Garhart, Casey .... 190 - 294
Garton, Roland 221
Gautreau, Ronald 168
Gilbert, Steven 260
Glasser, L. .... 180 - 297
Greenfield, Stephan 242
Greenhalgh, William .... 258 - 298

H **Page**

Hackler, Nadine 270
Halsey, Eric 281
Hamilton, Gregory C. 1
Hannafin, Michael J. .... 190 - 283 - 292
Harris, Albert L. 27
Harris, Jacqueline M. 27
Hayes, Joan 33
Heller, Marvin O. 277
Hemphill, Michael R. 195
Hirst, Steve 276
Howell, Jeff 273
Hughes, Curtis 283
Hyde, P. L. 96

J

Janta-Polczynski, J. 214
Jarom, Carol A. .... 120 - 131
Jensen, James D. 241
Jensh, Ronald P. 261
Jorczak, Robert L. 72

K

Kapp, Ann B. 271
Kearsley, Greg 199
Kent, Phyllis 272
Knechtel, John 205
Kurata, Masahiko 78
Kuznetsov, Helen 158

L

Laagland, Eelco F. 84
Lawson, Patricia A. 90
LeDoux, John C. 5
Leshin, Cindy 293
Levine, Lawrence 251
Lewis, Elaine 210
Lommel, Jane M. 245
Ludden, Laverne L. .... 136 - 275

M

MacLachlan, James 9
Malloy, Jane 13
Marshall, Ruth 52
Mayhew, Frances W. 53
Mellor, D. G. 96
Michael, Joel A. 43
Miller, John C. 252
Miller, Michael 275
Modell, Harold 263
Moehnke, Gary 242
Moore, John F. 284
Morariu, Janis .... 171 - 175
Morimoto, Yasuhiro 78
Morin, Patricia J. 264

PROCEEDINGS INDEX

N Page
Navarro, Aaron B. 288
Neuwirth, Sharyn 175
Newmen, Raye 240
Ng, Raymond 239
Novemsky, Lisa 253

O
Orchard, S. W. 180

P
Parker, Cindy 96
Phillips, C. B. 96
Phillips, Timothy 190
Pollock, Joellyn C. 293
Powell, Jan 36

R
Rambally, G. K. 214
Reed, Mary Jac 105
Reed, George 101 - 148
Reeves, Thomas C. 250 - 285
Reinhart, Richard 108
Reynolds, Jo Ann 265
Rhodes, Dent M. 267
Richards, Deborah E. 144
Richardson, W. Kirk 289 - 291
Rieber, Lloyd 190
Roskam, Ron 49
Rovick, Allen A. 43
Ryder, Randall 287

S
Sandals, Lauran 286
Sato, Takahiro 78
Saunders, Greg 49
Scherer, Chris 151
Scherer, Charlotte 39
Schroeder, Lois 272
Schulz, William E. 246
Shirey, Richard 279
Signer, Barbara 155
Singha, Raj 254
Smith, Lynn 221
Smith, Mary Grace 296
Smith, Philip 287
Sottneck, Henry W. 289
Standerfer, C. C. 195
Stanley, Charles J. 5
Stephens, Kenneth 17
Stewart, Tanis M. 110
Stoddard, Mary L. 227
Swigger, Kathleen M. 115

T Page
Teeter, Phyllis 287
Thomas, Barbara 49
Thomas, Donald H. 280
Thomas, Lynda 257
Thompson, Linda 257
Tisone, Mark J. 295
Trotter, Pete 276

U
Uppal, Arjinder S. 214

V
Vinall, Paige 131

W
Walla, Kay 255
Wedman, John J. 277
Weiner, Elizabeth 259 - 266 - 299
Wenger, Kathleen 184
Wesswick, Louise 58
White, Arthur L. 234
Williams, Allan 247
Wilson, J. H. 96
Wolfe, Wayne A. 278

Y
Yoder, Marianne E. 268
Young, Marsha 269

A Blueprint for CBT Courseware Development

Gregory C. Hamilton, PhD
Training and Development Department
The Timken Company
Canton, OH 44706

ABSTRACT

While the literature is replete with research reports, theoretical formulations, and speculations on the potential CBT offers, few papers present procedural information useful to those undertaking their first CBT project. This paper discusses the seven procedural steps of (1) defining terminology, (2) defining learner control options, (3) establishing standard display screens, (4) determining standard screen features, (5) using production teams, (6) developing a courseware production model, and (7) creating a planning model.

INTRODUCTION

The first steps in any curriculum development project are performed by using an instructional system design (ISD) model (Davis, Alexander, and Yelon, 1974; Esseff and Esseff, 1975; Gagne and Briggs, 1974; Kemp, 1977; Mager, 1975). To limit the scope of this paper, however, I'll assume you have followed those procedures and have decided to use computers as the delivery medium based on the compatibility between the computer and the subject matter.

This article outlines seven basic steps which, if adopted, will facilitate in developing computer-based instruction. To be sure, many aspects of a CBT project are evolutionary. However, addressing certain issues early on in a project can help reduce development time and minimize revisions.

DEFINE CBT TERMINOLOGY

One of the first tasks is to clearly define the terms that will be used throughout the project. Examples include terms such as simulation, feedback, question, practice, review, and summary. Although we all know what these terms mean, they tend to have personalized definitions often dependent on the context in which they're used. It becomes important to discuss and write down in clear, unambiguous language what the terms are, their meanings, and context for their use. This will facilitate communications between CBT project team members, assist in establishing a consistency to the CBT materials, and help to develop attendant cues for the student.

This last advantage is apparent when the term becomes associated with a particular CBT activity. For example, summaries could signal the end of the lesson, questions may appear only in testing situations, and practice could denote questions embedded in the instructional (non-test) material. The question designation might signal a multiple-choice response while practice and problems indicate free response and numeric input respectively. While good CBT should always include directions on how the student is to respond, explicit directions can be faded through the consistent use of terminology.

DEFINE LEARNER CONTROL OPTIONS

You will also need to establish and standardize control options which will be available to the learner. These options represent instructor defined functions that assist the learner through the courseware. As they are defined, you should reserve keys solely for their use. The ENTER or RETURN keys are commonly used by students to move forward. But you may want to allow the learners to page back

BEST COPY AVAILABLE

through the instruction as a review technique. Forward and backward movement is important; we do it with text books all the time. Other control functions may include referencing help, data, or facts tables, referring to a glossary, printing the screen, repeating simulations or animations, deleting input, etc. You should also decide how learners are to respond to system features such as menus, multiple-choice questions, and free response items. Once the options have been determined and the keys assigned, they should be used consistently in all lessons prepared. Do not use the ENTER key in one lesson and the RETURN key in another for the same purpose.

ESTABLISH STANDARD DISPLAY SCREENS

Screen function and design are crucial areas of initial planning for they become the basis for developing a style to your courseware. The use of standard displays like advanced organizers, a statement of the objectives, an option to redo selected portions or all of the lesson, and summaries are beneficial to both the learner and the developer. Standard use of beginning and ending displays also contribute to establishing a courseware style. Once standard displays have been decided upon, they should be used in every lesson you develop. This helps to establish not only a style but a consistency to the instruction. It also helps to reduce student confusion and anxiety. Standard displays serve as instructional roadsigns which are important to the learner, especially the novice. They also help in reducing development time since all authors follow the same format.

DETERMINE STANDARD SCREEN FEATURES

Most of your development time will be spent preparing the display screens. Several issues need to be resolved regarding the design and format of every display screen created. Headers indicating the lesson name or identifying the display type (menu, problem, feedback, etc.) assist in reminding the learners where they are and what they are supposed to do. Equally important are footers containing the lesson page number and available learner control options. The learner control options are often represented by small icons or mnemonic symbols. You need to experiment with different display formats and decide upon standard conventions that all developers will use.

There are several other aspects of screen design to be considered for the space between the header and footer. Text size and fonts, punctuation, use of color, and display order are a few such issues where a consensus among developers is needed. Some examples are: headers should always appear at the same location, in the same text size and font; the screen should be painted from top to bottom, left to right; graphics should appear first followed by text. Specific screen design guidelines are available (Alesandrini, 1984; Heines, 1983; Spannaus and Pariseau, 1984; Sweeters, 1985), however, personal styles need to be taken into account. A consensus, at the beginning of a project, upon one style will minimize time consuming revisions later.

USE PRODUCTION TEAMS

Many of the previous comments implied a "team approach" to instructional development. The principal team member is an instructional designer knowledgeable in instructional theories, motivational principles, curriculum development, and methods of interactivity and individualization. The second member of the team is the subject matter expert (SME). The success of any computer-based courseware is firmly rooted in good design and the quality of the information provided by the SME.

Other members of the team include a computer specialist and a graphic artist. While these four skill areas form the core of an effective development team, an extended "value added" team includes someone with formal training in the English language (human text editor) and someone skilled in statistical analysis, evaluation, and measurement techniques (psychometrician). While one person may possess several of these skills, it is very rare to find one person skilled in all areas.

Possibly more important than the composition of the team is communication among team members. Communication between the instructional designer and the SME is particularly crucial. But so are the links between the designer and the other team members. The computer expert must be able to explain hardware and software capabilities and limitations within the context of courseware design.

Linkage between artist, designer, and computer expert is required to design appropriate graphics that simultaneously support the instruction and are supported by the computer system. Communication between the designer and the language expert promotes efficient, concise, and unambiguous transfer of information to the student. Good message management is vital to the success of your instruction. Lastly, the interaction between the designer, SME, and psychometrician is important for it ensures that you will teach the student what you intend.

DEVELOP A COURSEWARE PRODUCTION MODEL

A production model specifies both the events and their sequence for courseware development. A simple three phase model could consist of (1) preparation, (2) pre-release, and (3) release components. The preparation phase, involving the entire production team, addresses the design, scripting, computer entry, and review of the instructional lesson. The pre-release phase focuses on the computer aspects of the lesson. Testing, debugging, and revisions occur in this phase. Forward and backward sequencing, learner control options, and input and response mechanisms should all function properly upon completion of this phase. This is also the time to check that the lesson does what it was designed to do, namely teach. The release phase involves a final review by the instructional designer and the SME and, of course, documentation. This model is similar to those used by other courseware developers (Bork, 1984; Hofstetter, 1984).

CREATE A PLANNING MODEL

The planning model deals with allocating human and physical resources, establishing schedules, and controlling costs. Typical management tools for such functions are PERT and Gantt charts, which graphically depict the chronological time frames for development activities. The value of these tools depends on the reliability of the input data, thus making the planning model empirical by nature. You need to keep track of all relevant development data (hours, costs, skill areas, etc.) broken down by activity and instructional strategy. Once the data has been recorded, standard anaylsis procedures can assist in determining the capabilities of your development team and setting reasonable time frames for CBT courseware production. While this model has to be continually updated as new and different instructional media/strategies are included, it holds great promise for CBT project managers.

The development of such a model must begin at the outset of any instructional development project. The point in the ISD system for using such a model is at the media selection stage. Given existing constraints of time, cost, and manpower, a functional planning model would be able to generate options for allocating limited resources. Avoiding frustration and disillusionment in any project, however, requires effective planning and management which, in turn, is based on information that accurately reflects the development team's capabilities.

SUMMARY

The purpose of this article was to present some fairly simple steps regarding the initial planning phases of a CBT development project. Seven key points were made.

1. Define terminology in clear, concise language.
2. Define learner control options.
3. Develop standardized display screens that appear in all CBT materials regardless of subject area or strategy.
4. Establish standardized screen features such as headers, footers, text sizes and fonts, display order, etc.
5. Use a production team for all phases of courseware development. At a minimum, the team should consist of an instructional designer, subject matter expert, graphic artist, and computer specialist.
6. Develop and use a production model specifying the development activities and their sequence
7. Create and maintain a planning model for management, documentation, and planning purposes.

While each of these steps represents a small portion of the total effort in preparing high quality CBT courseware, each has its merit. In all probability, anyone who has prepared computer-based instruction has recognized or actually completed these steps. If an explicit listing of these seven initial considerations assists you in your CBT programs, the article has served its purpose.

BEST COPY AVAILABLE

REFERENCES

Alesandrini, K.L. Five Star Screen Design. Data Training, 1984, 3(6), 39-41.

Bork, A. Producing Computer Based Learning Material at the Educational Technology Center. Journal of Computer Based Instruction, 1984, 11(3), 78-91.

Davis, R.H., Alexander, L.T., & Yelon, S.L. Learning System Design: An Approach to the Improvement of Instruction. New York: McGraw-Hill, Inc., 1974.

Esseff, P.J., & Esseff, M.S. Instructional Development Learning System. Columbia, MD: Educational Systems for the Future, 1975.

Gagne, R.M., & Briggs, L.J. Principles of Instructional Design. New York: Holt, Rinehart and Winston, Inc., 1974.

Heines, J.M. Screen Design Strategies for Computer-Aided Instruction. Bedford, MA: Digital Press, 1983.

Hofstetter, F.T. The Eighth Summative Report of the Office of Computer-Based Instruction. University of Delaware, 1983.

Kemp, J.E. Instructional Design: A Plan for Unit and Course Development (2nd. ed.). Belmont, CA: Fearon, 1977.

Mager, R.F. Preparing Instructional Objectives (2nd. ed.). Belmont, CA: Fearon, 1975.

Spannaus, T., & Pariseau, C. Aesthetics and Instruction. Data Training, 1984, 3(4), 41-45.

Sweeters, W.G. Screen Design Guidelines. ADCIS 26th International Conference Proceedings, 1985, 42-45.

Making Feedback in Computer-Based Training Meaningful

John C. LeDoux

Charles J. Stanley

FBI Academy
Quantico, Virginia

ABSTRACT

Feedback provided in computer based training (CBT) lessons assumes the role of a patient and helpful mentor for the adult learner. Meaningful feedback assists a student in assessing the knowledge gained, thus aiding the learning process. Two general principles and several specific guidelines are proposed to help CBT authors in designing appropriate feedback.

INTRODUCTION

Various aspects of adult learning theory suggest that well designed computer-based training (CBT) may be an ideal way to provide certain learning opportunities for adults. First, adults who participate in CBT will feel a greater sense of control in the learner role than will those who listen to a lecture (LeDoux and Stanley, 1985). Second, the flexibility and efficiency of CBT should appeal to the time orientation of a mature student. And third, a well designed CBT course provides a student with the opportunity to concentrate on those lesson segments that are most personally significant (Reynolds and Samuels-Campbell, 1985.)

Given these considerations, it would seem that CBT can't fail. Of course it can, and frequently does (Steiner and Graver, 1984). While there are several factors that can precipitate failure, a lack of meaningful interaction seems to be one of the most prevalent (Zemke, 1984). In CBT, meaningful interaction implies a learning environment in which the student actively responds to the information presented. In turn, the computer analyzes the responses and provides comments, "a process more commonly referred to as feedback" (Jonassen, 1985, p. 7).

Meaningful feedback assists a student in assessing the knowledge gained, thus aiding the learning process. Learners must be aware what they do know and what they don't know, as well as why they are correct or incorrect. In CBT, feedback assumes the role of a patient and helpful mentor. The purpose of this article is to suggest two general principles and several specific guidelines that can be applied in CBT lesson design.

GENERAL PRINCIPLES

We may not view CBT lessons as being tests, but an important element of CBT involves monitoring the learning process. A person is provided information, presented with a question or problem, and asked to make a decision. Based on the student's response, appropriate feedback assists in self-evaluating mastery of the subject matter.

More thought must be given to feedback in CBT than in a traditional testing situation. With typical classroom tests, a student selects answers and is later told which are incorrect. A good instructor will give the class the opportunity to ask questions about the test. There is seldom time, however, to give each student specific feedback on each question.

This is an area where CBT has the potential to be different from, and superior to, traditional testing situations. A student has the opportunity to obtain specific meaningful feedback for every response. Feedback is not just a terminal aspect of testing, but an integral part of the immediate interactive learning strategy.

The more limited students are in terms of the responses they may make, the less difficult it is to supply appropriate feedback. Authors will find it easier, for instance, to design feedback for multiple choice questions than for short answer questions. The latter possess a wider universe of possible answers. Nevertheless, one must try to anticipate in advance ALL the possible responses that can be given and the rationale underlying the responses. Depending on the question type, authors must prepare for four categories of answers: CORRECT, INCORRECT, UNANTICIPATED, and INVALID.

CORRECT answers are acceptable answers specified by the course author. The various answers are synonyms or will differ with regards to such elements as spelling, punctuation, or capitalization. All, however, are still considered basically correct. For example, the answers 'Malcolm Knowles' or 'Malcolm S. Knowles' may be CORRECT answers to one question as might be 'car' and 'auto' to another. By careful quality control the author should be able to ensure that most possible CORRECT answers have been specified.

The principal challenge is to craft feedback that captures the numerous wrong answers a student might give. Studied consideration of potential INCORRECT answers is mandatory. Feedback should not only tell a student that he or she is wrong, but also should explain why. All anticipated INCORRECT answers must be incorporated into the branching design, if possible, so a student will have the opportunity to receive specific feedback that explains an error. Otherwise the learner may receive the far less specific feedback designed for UNANTICIPATED answers.

UNANTICIPATED answers are those responses that the author did not consider when designing feedback. The response may be correct or incorrect, yet unexpected. For example, an answer of 'M. Knowles' in the first example above would be correct, but the lesson was not set up to recognize it as such. Likewise, an incorrect answer of 'Bart Starr' would not be anticipated to a question dealing with adult education. Since response correctness is unknown, one acceptable feedback strategy is to acknowledge the response is not one expected and direct the student to try again. Hints or additional instructions may be given to guide a student.

INVALID answers are those that do not make sense, such as a response of "E" to a multiple choice question that has answer possibilities "A" through "D". Within the constraints of the development system, course authors may choose to either deactivate those keys that may not be used or supply feedback that clearly states the answer is not capable of being judged. This response category may be most applicable to forced choice questions.

GUIDELINES FOR FEEDBACK

The key aspect of CBT is the interaction between student and computer. Thus, provide for as much interaction as possible. The proactive learner is more likely to learn than the passive recipient of a lecture (LeDoux and Stanley, 1985). Therefore, supply interaction opportunities at least every three or four screens or, alternatively, about once per minute. "PRESS SPACE BAR TO CONTINUE does not provide for significant interaction (Heines, 1985) and we know of nothing more potentially monotonous in a CBT lesson.

The term 'screen' requires some elaboration. We count each occasion in which the learner overtly responds to information as one 'screen'. Thus, a short series of screens that uses a time delay strategy may be considered one screen. Regardless of the criteria used, authors should count 'screens' from the learner's viewpoint.

To force interaction is as equally undesirable as it is to ignore it. Avoid asking questions that are inconsequential or that have obvious answers, since these require limited student thought. Feedback for these questions is bound to be trivial.

Feedback should advise concisely if the answer given was correct. Initial words such as "Correct" or "That's right" will serve well. On the other hand, comments such as "Super" or "Fantastic, you really know your stuff" are fawning and suggest a significant accomplishment. If an adult had to guess three times before getting the right answer that person is not likely to feel that it was 'fantastic'.

Likewise, if the student is wrong, so advise. Some authors prefer a generic comment such as "You didn't pick the best answer." Generic responses should be limited to topics such as those dealing with human relations where there may be few definitive answers. One reason the generic answer is popular is simply because it is generic. We have observed lessons that first tell the learner the best answer wasn't picked and then urges another try. Through the use of this vague feedback the author has avoided the exacting task of designing meaningful feedback to anticipated incorrect responses.

Feedback must do more than advise a student as to correctness. To indicate simply that the learner is wrong is a less effective form of feedback (Noonan, 1984). We believe that, in the case of incorrect responses, a student will often misapply fact(s) to reach a wrong conclusion. That is, an adult learner will miss questions not due to total ignorance but, rather, due to partial knowledge.

Based on this perspective, there are four potential parts to feedback for an incorrect response: 1) advising that the response was incorrect, 2) indicating what elements of knowledge concerning the question or problem the learner seemed to have, 3) suggesting where the logic applied in making the response went astray, and 4) supplying the correct answer (if the learner will not be given the opportunity to try again).

Feedback for correct answers must be designed with similar care. Merely advising a student that he or she is correct is not enough. The reason why the answer was correct should be given as well (Cramer, 1984). Feedback must take into account the possibility that the correct response was obtained by chance. Even if guessing was not involved, the right answer may have been given for the wrong reason.

Standardize the location and format of feedback. A student should be able to anticipate the location of the feedback. Establish a functional area on the screen where feedback is to appear (Sweeters, 1985). Of course, some flexibility in the screen design is allowable depending on the amount of information that must be displayed, but relative consistency is the key.

The question type determines the number of attempts at answering given a student. For short answer questions that have numerous alternative responses, two or three attempts should be permitted. Two attempts should suffice for multiple choice questions or for each item of a matching question. Dichotomous questions should be limited to one attempt since missing the question makes the correct answer obvious.

CONCLUSION

A CBT author must give thoughtful consideration to feedback when designing lessons. Carefully crafted feedback fosters a meaningful learning experience and is essential if CBT is to be truly 'interactive'. Hasty, ill considered feedback destroys the benefits of CBT that may be enjoyed by the mature learner. Naturally, the questions or tasks initially eliciting a learner response must be equally well designed. The successful CBT author will ensure that the composite interactive elements are well constructed and relevant.

REFERENCES

Cramer, S. E. The instructional value of wrong answers. Paper presented at the 1984 MUG/USG Conference, Atlanta, Sept 1984.

Heines, J. M. Interactive means active. Learner involvement in CBT. Data Training, 1985, 4(4), 48-53.

Jonassen, D. H. Interactive lesson designs: A taxonomy. Educational Technology, 1985, 25(6), 7-16.

LeDoux, J. C. & Stanley, C. J. A comparative study of computer-based instruction vs. lecture. Proceedings of the 26th International Conference of the Association for the Development of Computer-Based Instructional Systems, 1985, 10-14.

Noonan, J. V. Feedback procedures in computer - assisted instruction: Knowledge of correct response, process explanations, and second attempts after errors. (Doctoral dissertation, University of Illinois at Urbana-Champaign, 1984). Dissertation Abstracts International, 1984, 45,01-A.

Reynolds, A. & Samuels-Campbell, C. Simulations: Time to take another look. Performance & Instruction, 1985, 24(4), 15-17.

Steiner, H. & Graver, J. What's wrong with CBT? Now, before it's too late. Data Training, 1984, 3(4), 27.

Sweeters, W. G. Screen design guidelines. Proceedings of the 26th International Conference of the Association for the Development of Computer-Based Instructional Systems, 1985, 34-37.

Zemke, R. Evaluating computer-assisted instruction: The good, the bad, and the why. Training, 1984 (May), 22-47.

ADVERTISING SPONSORSHIP OF SOFTWARE:

WHO COULD BENEFIT?

James MacLachlan, Associate Professor

Rensselaer Polytechnic Institute, Troy, NY 12181

Abstract

The costs of producing and broadcasting television entertainment are paid by advertisers in return for the opportunity of presenting commercial messages to the audience delivered by that medium. In like manner, it would be possible for the costs of producing and disseminating certain computer based tutorial programs to be borne by advertisers in exchange for opportunities to present commercial messages to the audiences delivered by those tutorials.

The economics of this are examined from the point-of-view of advertisers, and it appears that computer based tutorials could deliver "reminder" advertising effectively to certain audiences. However, there are two possible detrimental effects: (1) software efficiency may be reduced by time taken up by advertising messages, and; (2) students may be encouraged to use products which are not in their best interests. Ways of minimizing such detrimentally effects are suggested.

Advertising Sponsorship of Software: Who Could Benefit?

Schools are lamenting a lack of affordable software for their personal computers. One possible solution to the problem might be free software paid for by advertising sponsorship. We will examine this possibility from two perspectives:

1. Would it be beneficial to advertisers to sponsor software?
2. Might advertising have a detrimental effect on students?

Would Advertisers Benefit?

This article will present general principles concerning advertising effectiveness. As each principle is presented, we will consider how it impacts the following advertising objective and media possibility:

> ADVERTISING OBJECTIVE: Kellogg's wishes to have the public "Energize at Sunrise" with a good breakfast. Many people do not eat breakfast, and Kellogg's knows that if they would eat breakfast Kellogg's products would be chosen in many instances.
>
> MEDIA POSSIBILITY: A spelling tutor program suitable for junior high and high school use. The program conducts a spelling bee in which one or more students may participate.

Methods for evaluating the effectiveness of advertising vehicles are well established (Sissors and Petray, 1976), and we can expect that the advertising opportunity presented by sponsorship of software such as computerized tutorials will be evaluated according to established criteria regarding these concerns:

Target Audience

No advertising media is useful unless it delivers a target audience of interest to the sponsor. This audience is usually defined in these terms (Kotler, 1984):

- Demographics: such as age, sex, family size, income, occupation, and education
- Geographics: such as the region of the country lived in, climate, and urban/rural
- Psychographics: such as social class, lifestyle and personality
- Behavioral: the extent to which they use the product, the benefits sought, and brand loyalty.

To relate this to the example, Kellogg's target audience for the "Energize at Sunrise" message would include junior and senior high students. Even though they may not do much of the actual purchasing of the cereals, they are potential consumers and more cereals will be bought by parents as they consume it.

Message Considerations

There are generally considered to be three basic types of advertising approaches:

1. Information oriented advertising presents sufficient facts so consumers can visualize how the product will benefit their lives. Information is usually cast in terms of benefits the consumer will derive, rather than descriptions of products, because the assumption is that consumers are buying benefits rather than products per se (Gallup, 1974). Messages may be quite lengthy.

2. Image oriented advertising uses analogies and similes to position the product in the consumer's mind (Gilson and Berkman, 1980). Frequently, the product is characterized in terms of who uses it (successful, famous, attractive) and the circumstances (fun, enjoyment, excitement).

3. Reminder advertising attempts to maintain high brand and product awareness. The general assumption is that share of market is a function of "share of mind", which is in turn a function of "share of voice" in the cluttered world of advertising. The reminder may be to purchase the product, prepare the product for later use (eg. Jello), or to consume the product (eg. eat cereal for breakfast).

The purchase process itself is characterized by hierarchical models, such as the "Path to Purchase" model (Lavidge and Steiner, 1961), which assumes that buyers pass through these stages:

- unaware of product
- aware
- knowledge about product
- liking
- preference for this product
- conviction that purchase would be wise
- purchase

Advertising facilitates passage from step to step in this hierarchy. Information oriented advertising is most successful when the prospect is at the first three stages (unaware, aware, knowledgeable). Image advertising is most useful when the prospect is in the liking, preference and conviction stages because in these stages he or she must develop a personal liking for the product. Reminder advertising is most effective when the prospect has completed other steps in the hierarchy and must be motivated to make the actual purchase.

As regards the "Energize at Sunrise" example, reminder advertising would be appropriate because most prospects already have favorable attitudes toward the product and need to be encouraged to consume it on a regular basis. Frequent and short messages would be successful in accomplishing this objective. Such reminder messages could be imbedded in a tutorial at one or more of the following junctures:

- The initial startup screen could include a slogan such as "This program has been distributed free of charge by Kellogg's who remind you to 'Energize at Sunrise' with a good breakfast." Several different versions of the message could be developed and one would be randomly selected when the program is booted.

- Header or footer lines on the screen could include the Kellogg's name or the slogan. For example, it could be imbedded in the screen number status line at the bottom of the screen.

- Wait screens could display the message while disk access or computations were being accomplished, but this would be undesirable because the unpleasant experience of waiting for the computer would be associated with the Kellogg's name.

- Commercial message screens could appear at various times with information such as: "Students who eat good breakfasts get better grades."

All except the last tactic would not intrude on the user's enjoyment of the program, yet they would enable reminder messages to be presented effectively.

Cost Justification Criteria of Advertisers

Advertisers are concerned about reaching their target audience at a low cost per thousand exposures (CPM). It is the responsibility of the media to document that they deliver a given audience; the advertiser then divides that audience (in thousands) by the cost of the media buy to get the CPM.

In the case of television, the networks pay the A.C. Nielsen company to estimate the audience for each of their shows by using a sampling procedure. Certain homes are equipped with black boxes called "Audimeters" on their TV sets which record the programs watched on the set and relay that information by phone to A.C. Nielsen who then analyse and tabulate it using computers.

Magazines and newspapers report the number of subscribers and newsstand sales and have that information verified by the Audit Bureau of Circulation.

It would be difficult to audit the use of computer software with the same levels of precision. Questionnaires could be sent to teachers to find out the extent to which they are using certain sponsored software. They should be asked to estimate the number of students using those programs and how frequently the typical student uses each program during the course of a school year.

The cost of distributing computer software is quite low. One disk would be sufficient for each school because it could be copied as many times as desired, although additional disks might be needed if more than one type of computer is used at a school. Documentation can be included on the disk to reduce documentation costs, as is the case with the personally developed software written by IBM employees which is marketed by IBM. Alternatively, the documentation could be quite elaborate and reinforce the program contents with pictures and charts. Such documentation would provide another opportunity for presentation of advertiser messages. Also, an advertiser might audit the success of a program by noting the number of requests for the documentation, especially if the documentation is of workbook form where the teacher should request one for each student.

Perhaps the main cost would be developing the software itself. These costs, however, are being reduced by improved software, such as PILOT, which facilitate the development of tutorials on PC's. Also, as more and more schools acquire computers, these development costs will be amortized over a larger base.

Would There Be a Detrimental Effect on Students?

There are two main possibilities for a detrimental effect:

1. Software efficiency may be reduced by time taken up by advertising messages, or, while the program is in progress students may be distracted by embedded advertising.

2. Students may be encouraged to buy or use products which are not in their best interests.

Advertisers can eliminate the first possible problem by not undertaking any activities which would slow the execution of a program. This would imply that advertiser messages of any length be displayed in the startup screen only. Advertiser ID's while a program is running, if any, should be subtle so as not to distract attention from the learning purpose of the tutorial.

Concerning the second potential problem, messages can be grouped into three general categories:

- **Beneficial**--Those that recommend conduct widely considered to be desirable, such as saving money or eating a nutritious breakfast.

- **Neutral**--A recommendation to buy a parity product, such as a particular brand of pen. It is not very important whether students use one brand of pen or another.

- **Detrimental**--A recommendation to buy or use a product which could be detrimental to health or well being, such as "Junk foods."

Probably school officials would have no problems with products in the first or even second category. The third category could cause repercussions. For example, mothers might object if the computer were telling their children to "Drink Coca Cola" because of the sugar and caffeine content of that beverage. Such advertisers would be wise to avoid this medium, because negative publicity could result.

School officials will need to know exactly what advertising messages may be included in a program so they can make a judgment as to whether it would be appropriate for their pupils. In order to facilitate this, advertisers should disclose complete details of embedded commercial messages, doing this both in a text file on the disk and in the documentation. It also should be described on any catalog sheets or request forms.

It will not be possible for school officials to run a program to examine any embedded advertising. This is both because the process would be time consuming, and because certain advertising messages might appear on a random basis and not be seen during the time of administrator examination. For this reason, advertisers must disclose their advertising material with complete candor for school administrator scrutiny.

If proper attention is given to the concerns described here, there should be no detrimental effect on students.

Conclusions

In the past, advertisers have sponsored 16mm films for educational use, and it is thought that both the advertisers and students have benefited from that practice. In like manner, computerized tutorials can provide a suitable vehicle for certain advertisers, particularly those wishing to direct reminder advertising to a student audience. If the restraints and controls suggested here are exercised, the result can be mutually beneficial.

Advertiser sponsorship of computerized tutorials could become a widely accepted practice and result in much useful free software. As time goes on, the practice could become more and more common because computers will be more widely available; and, when optically readable data disks become common for personal computers, a wealth of free software might be distributed on a single disk greatly reducing the costs of distributing free software. The ramifications for advertisers and the public are substantial.

REFERENCES

Gallup, George (June 1974). "How Advertising Works," Journal of Advertising Research, Vol. 14, No. 3.

Gilson, Christopher and Harold W. Berkman (1980). Advertising, Concepts and Strategies. Random House Business Division, New York.

Kotler, Philip (1984). Marketing Management: Analysis, Planning and Control, Fifth Edition. Prentice- Hall, Englewood Cliffs, New Jersey.

Lavidge, Robert J. and Gary A. Steiner (October 1961). "A model for Predictive Measurements of Advertising Effectiveness," Journal of Marketing, 59-62.

Sissors, Jack Z. and E. Reynold Petray (1976). Advertising Media Planning. Chain Books, Chicago.

MICROCOMPUTERS IN BUSINESS PROGRAM

Jane A. Malloy, Ph.D.

Delaware County Community College

This innovative microcomputer curriculum appeals to different types of students who are interested in learning about the microcomputer for business, home or office use. The courses include computer terminology, word processing, graphics, spreadsheet analysis, data base management, BASIC, COBAL, and PASCAL. Students who statisfactorily complete the 15-credit program are awarded a certificate of competency in Microcomputer Use for Business. Students may also elect to take individual courses.

* * * * * *

PC's In The PM is an innovative curriculum which was recently instituted at Delaware County Community College to meet the current business and personal demand for more knowledge about microcomputers. The success of the Microcomputers in Business Program hinges on its appeal to evening and weekend students who are interested in acquiring expertise with the microcomputer. The students attain a basic knowledge of microcomputers, their applications, selection, capabilities and limitations, as well as instruction about and pertinent use of the most popular software packages.

Individuals may work toward the 15-credit certificate of competency or just take courses for personal enhancement. It should be noted that all courses consist of "hands-on" workshops and classroom lectures. Therefore, these courses provide the student with an opportunity to acquire greater knowledge about the operation and software packages available to the microcomputer than those provided in a typical training program. Also, the student has time for programming with specific and self-chosen assignments.

Upon satsifactory completion of the certificate program, the student should be able to:

..... Explain the purpose of major hardware components.
..... Define common terms applied to microcomputers.
..... Write programs in the BASIC language.
..... Evaluate hardware and software for use in a small business.
..... Use disk drives and printers associated with a microcomputer.
..... Demonstrate an understanding of some other facets of microcomputer applications (successful completion of electives).

CORE CURRICULUM

This consists of five required courses and three electives. The classes typically meet two nights per week for 2 1/2 hours and two credits. The students receive letter grades for all of the classes. The first class listed, Introduction to Microcomputers, is a prerequisite for many of the other courses. In addition, Microcomputer Programming Concepts is a prerequisite for the

BEST COPY AVAILABLE

computer language courses. The prerequisite requirement can be met by, of course, taking the required course or by taking a placement examination or by permission of the instructor.

TITLE	CREDIT
Introduction to Microcomputers	2
Microcomputer Operating Systems*	2
Microcomputer Programming Concepts*	2
Evaluating Micro Systems*	2
Microcomputer Programming in BASIC*	2
Microcomputer Electives	5
TOTAL	15

ELECTIVE CURRICULUM

The following are a representative list of the microcomputer electives which are offered:

Microcomputer Communications*	1
Graphics Usage in Microcomputers*	1
Microcomputer Spreadsheet Software*	2
Data Base Management Systems*	2
Microcomputer Word Processing Orientation*	1
Logo Applications in Education	2
Microprogramming (PASCAL)*	2
Microprogramming (COBAL)*	2

*Require prerequisite

CORE COURSE

The Introduction to Microcomputer course is taken by all students who have no prior knowledge of computers. As noted previously, the students are able to omit this course if they take a placement examination or if the instructor gives permission. This nontechnical introduction is designed to provide students with a basic understanding of microcomputer systems, hardware, software, and information processing using computers. Included in this course are five broad topics which are covered on a weekly basis. These topics encompass computer terms and its keyboard, disk operating system, word processing, data base management, and the spreadsheet analysis.

Lecture

The course typically begins with a lecture/discussion of simple terms which include pixels, RAM, ROM, default, modems, disks, disk drives, configuration, etc. Often, the next step in this course is "hands on" where the students learn the keyboard. Usually, the students will have some knowledge of typing; so, only the unique microcomputer keys are stressed. Several simple exercises are given the students to complete which utilize these keys. Usually after the students have been given the opportunity to use the microcomputer, they feel much less intimidated by the equipment.

Disk Operating System

Next, the student is introduced to the Disk Operating system--its purpose and use. Some of the commands which the students perform on the microcomputer are formatting a disk, copying a disk from one drive to another, and deleting files.

Word Processing

The word processing aspect of the course is designed to offer the students an opportunity to learn a software package for their personal use. This word processor will allow them to manipulate characters, words, sentences and pages to create a document that fits their needs. However, within the time limitations of this course, the students will be able to initially enter short paragraphs and revise them and progress to being able to input and revise a letter.

At the conclusion of the word processing course segment, students are usually aware of the advantages of a word processing package over a typewriter. To enhance this view, the students are encouraged to bring a personal document that needs to be written and they may use it as a class exercise. Often the students are able to print a document that would satisfactorily meet the typing and placement standards of any office.

Data Base Management

It should be remembered that word processing is considered primarily responsible for placing microcomputers in offices around the world because it is basically an easy concept to understand. The resultant increase in office productivity justifies the purchase of a computer system. Once people become comfortable storing documents and other information in their computer, they usually wish to maintain their records in a more orderly fashion. Data base management affords the student the opportunity to manipulate files, records, and fields; one is able to manage data anyway one wishes (Green, 1983, p.2).

Students are able to discuss the general concepts of data base management and design a data base for a particular application upon completion of this segment of the class. In addition students can access various records from the data base and produce a standard report.

Spreadsheet Analysis

The last segment of the class is devoted to spreadsheet analysis. The most popular software program is used--Lotus 1-2-3. Given the time constraints of the course, the students are able to resolve a number of business problems with this package, such as a simple balance sheet and income statement. The students learn how to manipulate the columns and rows in order to set up various spreadsheet formats. Standard mathe-matical functions are also included. By the end of this application, the studens design a spread-sheet for a particular business use.

SAMPLE ELECTIVE COURSE--GRAPHICS

Since a major use of graphics in this area is for business applications, the course is set up so that the students will be able to use this feature on the job. At present the course is divided into two segments: graphics using the BASIC language and graphics using the software package, Lotus 1-2-3.

Language

It is, of course, assumed that all students will have some familiarity with BASIC and microcomputers before taking the course since this class requires a prerequisite. The students begin with some fundamental graphic concepts, such as the selection commands for background and foreground colors, line, draw, pset, etc. The students enjoy plotting lines, rectangles, circles and ultimately having them interact with animation. One of the examples is termed "jaws" and it is intriguing to watch the lines connect and the colors change as a result of the students' BASIC programming. This programming is guided by the instructor and/or text and the students are encouraged to adapt it to their needs.

The students need no reinforcement to utilize the business applications because they are all well aware that "a picture is worth a thousand words", particularly when one is trying to describe a complex business problem. The students learn how to design all different types of charts and label them on both axes, as well as give them a title. They are also aware of the need to draw them to scale so that the graph does not distort the information that is being presented. In other words, there are limitations to the graphics that one is able to produce; the number of pixels is a determinant. If the chart numbers are larger than the number of pixels, it will have to be drawn to a specific scale.

Spreadsheet

In order to use the graphic capability of Lotus 1-2-3, one has to first create a spreadsheet. After this is done, the software program is able to create graphics using specific directions that one keys into the computer. This package creates a line, bar, stacked-bar, pie or xy charts according to your selection. A nice feature of this package is that it enables one to view the graph at any time and display up to six different ranges. A different crosshatching pattern is used for each of the data ranges since they are usually printed in black and white. However, it is possible to have them displayed in color. The students learn how to label their graphs and give them a title; it is also possible to include a legend describing the crosshatching pattern with the graphs. The students conclude the course by printing the graphs.

SUMMARY

PC's in the PM is a new approach to microcomputer training. The students enjoy the openness of the curriculum whereby they can take as many courses as they wish or the entire 15-credit microcomputer curriculum. Often, businesses send their employees so they will be able to successfully operate desk top computers. Occasionally, students elect the classes because they have bought a computer and cannot operate it or they are considering purchasing one. The curriculum is flexible and responsive to new technology, therefore, classes can be added or restructured. This curriculum is in the forefront of the community demand for knowldege about microcomputers.

REFERENCES

Bitter Gary G. and Camuse, Ruth A. (1984). *Using a Microcomputer in the Classroom*. Reston: Prentice-Hall, Inc.

Chambers, Jack A. and Sprecher, Jerry W. (1983). *Computer-assisted Instruction*. Englewood Cliffs: Prentice-Hall, Inc.

Getting Down to Business with Graphics (1985). *Computerworld*, 19, (15A).

Green, Adam B. (1983). *dBase II User's Guide*. Englewood Cliffs: Prentice-Hall, Inc.

Huntington, J. F. (1980). Microcomputers and University Teaching. *Improving College and University Teaching*, 28(2), 75-77.

Meirhenry, W. C. (1982). Microcomputers and Adult Learning. *Training and Development Journal*, 36(12) 58-62, 64-66.

Self, Charles C. (1983). A Position on a Computer Literacy Course. *Resources in Education*. November, 1-39.

STATE NOTATION AND A TOP-DOWN INSTRUCTIONAL DESIGN METHODOLOGY

Kenneth Stephens

ManTech Services Corporation

ABSTRACT

State notation is a method of documenting the design of real-time software, like computer-assisted instruction (CAI) courseware. It lends itself especially well to a structured design methodology. The rules of a hierarchical version of state notation, and the way it is used in a method of progressive refinement of lessons, will be described.

Much of the process of designing courseware is still at the state of an art, rather than a science. Lesson design, as practiced by many designers, is still an unstructured process. Too often, lessons are allowed to simply evolve from a few central ideas, loosely tied together. A step above this level of chaos is the use of a single gigantic flowchart, which grows to occupy an entire wall in the designer's office. It quickly becomes too unwieldy to work with, and difficult to modify.

There are parallels here between courseware development and other types of software development. It has long been recognized in the software engineering world that there are advantages to be gained from structured methods and modular development. Anyone who has had the unfortunate experience of trying to read through "pretzel code" (either someone else's, or your own a few months after the initial writing) to find a bug or to add enhancements will surely agree. There is a developmental singularity that results from an approach that progressively refines a problem, and postpones implementation decisions until the appropriate time. Wirth (1971) introduced some of these notions of "top-down" design and the method of progressive refinement. Among the recommendations that was made in that paper was that "a notation which is natural to the problem in hand should be used as long as possible...". Many different types of notation have been suggested to describe the behavior of software, and perhaps each has its own merits, depending on the type of problem (Belady, Evangelisti, and Power, 1980).

The present paper will describe the use of state notation in a system of structured instructional design, and argue that it is a notation that is natural to the problem of courseware development and implementation. It is, unfortunately, not as widely known outside of engineering circles as is flowcharting. State notation was developed in the mid-1950's to describe and predict the behavior of electro-mechanical switching circuits (Mealy, 1955; Moore, 1956). It had the advantage that events in real time could be used as inputs, conceptually driving the system through a series of states in a predictable manner. Not long after this initial application, state notation was applied to other kinds of sequential processes. In the operant conditioning literature, state notation was applied to the task of electronic programming of schedules of reinforcement (Kushner, Knapp and Snapper, 1967). The SKED language for laboratory computers (registered trademark of State Systems, Inc., Kalamazoo, MI) evolved to successfully meet these needs in the research psychology community (Snapper, Kadden, and Inglis, 1982; Stephens and van Haaren, 1977). Other kinds of transition networks are formally very similar to state notation (Woods, 1970; Kieras and Polson, 1982; Dwyer, 1981).

There are three compelling reasons why the experience of the psychological laboratory is relevant to instructional technology. First, computer-assisted instruction (CAI) lessons and the laboratory control software are computer programs designed to implement real-time procedures. Even though the nature of the behavior that serves as input to the program and the nature of the stimulus conditions produced as a result are considerably different, both CAI and laboratory programs written in a language like SKED can be used to train new performances. Instruction is a shaping procedure, in which verbal responses are to be added to the student's repertoire. A second reason for the use of state notation is conceptual. It may be helpful for the designer to conceive of a student repertoire as existing in a number of discrete states throughout the course of the lesson. The objective of the program is to facilitate the transition from an initial entry state to a goal state that corresponds to mastery of all the instructional objectives along the way. A state diagram of the procedure helps to visualize the transitions and the student behaviors that enable them. The third reason, to be clarified in a later section, is the compatibility of state notation with a top-down, progressive refinement methodology for instructional design like that advocated above.

BASIC RULES OF STATE NOTATION

A state is associated with a set of properties or attributes. It can be distinguished by the stimulus conditions or displays, or by the response options available, or by the logical path through the lesson which can follow. States are represented as circles in a state diagram. A collection of states is called a state set. Within a state set, one and only one state can be active at any given time. Certain events, either caused by student responses or the passage of time, serve as inputs and cause transition between states. The kinds of input events will depend on the level of refinement at a particular stage of the design process; the concept of progressive refinement will be discussed later. The screen displays, video sequences, and audio that may be presented to the student represent outputs produced when a new state becomes active. The state transitions are represented as arrows pointing from the prior state to the resultant state. Whatever event causes the transition is labeled on the arrow, or transition vector. State transition is considered to be instantaneous at the moment the input requirements for that transition are met. It is possible for an unlimited number of events to have the potential to cause transition from one state to others, although in practice, each state usually has no more than four or five. With multiple options, the transition vector whose input requirements are satisfied first will be the one that determines which state is active next.

Transitions are of three types. Transitions from one state to the next have been previously described. Any response requirements on the transition vectors for the new state are all reset. When there are transitions from a state back into itself, (the vector loops out of the circle and curves back into the same circle) the same state remains active but all the input requirements associated with the transition vectors of that state are reset. It is as if the state were entered for the first time. A third type of transition is not really a transition at all and is called a pseudo-transition, or null transition. This is useful when an event is to be noted, and perhaps a flag is set for later use, but the current state doesn't really need to change. This type of transition is notated as a vector to the letters "SX" (not another circle). These types of transition are illustrated in Figure 1.

A few other conventions have been added to state notation to make it easier to use and to increase its power. For instance, decision functions (IF-THEN-ELSE rules) coupled with the use of variables, and probability gates (WITH P=xxx) have been used in the SKED language and can also be used in a complex instructional procedure without violating any of the rules, by inserting a decision diamond (just like flowcharting) on the transition vector out of a state.

Up to this point, we have been talking about processes that are simple enough to be completely described using a single state set. With more complex processes or when there are a number of processes operating concurrently, it is advantageous to describe the procedure with a set of parallel state sets. The modularity that results makes complexity much more manageable. Another reason to make use of parallel state sets is to enable a structured breakdown of a problem, such as that in a systems approach to instructional design and

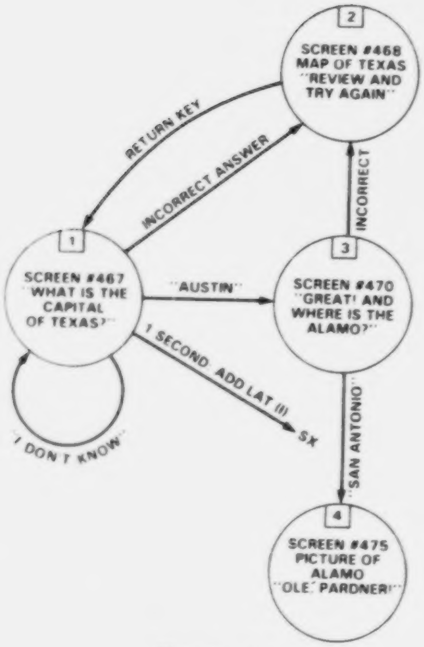


Figure 1.

implementation. Imagine a hierarchy of state sets, in which the grossest relationships are portrayed in a high-level state set, showing the flow of control between major concepts or tasks. Each state of the top-level state set is more fully defined in another state set at the next lower level of the hierarchy. There might be a whole family of second-level state sets, each corresponding to one state of the top-level structure. Each state in a particular second-level state set has the status of a subtask or subconcept, and has its own state set at the third level, and so on, until the states of the lowest-order state set specify all the detail needed to program the courseware. These bottom level states correspond to "frames" in the lesson. In practice, there is too much to specify about a frame to fit into the circles and arrows of a state set, so it is convenient to reference an external form that specifies the screen display, any video or audio, and the consequences for various anticipated student responses. The bottom-level state set shows the flow of control among the set of frames in a small instructional sequence for a minimal-size concept or task, and higher-order state sets show a different level of flow of control. The transition vectors at the lowest level are usually specific to certain student responses, while the arrows in higher state sets refer to completion of lower-level subprocedures, or mastery of objectives.

When there are parallel state sets, there has to be some mechanism for synchronizing the actions of the parallel processes. In earlier implementations of state notations (Snapper, Stephens, and Lee, 1974), this was done by way of a "pseudo-event" called a Z-pulse which was generated as an output upon state transition in one state set, and was immediately made available to all other state sets in case the currently active state of one or more of them could use the Z-pulse as an input event, triggering another state transition in that state set. Conceivably, a hierarchical scheme could be implemented just by letting the transition events of a high-order state set be Z-pulses that would be produced upon completion of the processes in a lower-ordered state set (the Z-pulse at the end of one subprocess would be the start event for the next subprocess in the main process). This notation, while adequate and occasionally powerful, is too unwieldy for a hierarchical system. We will retain the notion of Z-pulse events, but not rely on it. Instead, we will adopt some new conventions that will make state notation more amenable to the hierarchical approach. First of all, in existing versions of state notation, all parallel state sets are always active, although they may be sitting in an initial state waiting for an event that will trigger the activity they were designed for. In a hierarchical scheme, it is convenient to introduce the notion that a state set is inactive unless the higher-order state to which it corresponds is active. Usually, entry into a state in the higher-order state set will activate the lower-order state set corresponding to it. The high-order state will be in effect until the low-order state set has reached its terminal state, and this will be the event that triggers state transition at the upper level. However, there may be other events at the upper level that cause state transition. In this case, the default is that the lower order state set is deactivated and the next time it is activated, its initial state will be again active. To denote this contingency, the word RESET can be listed as an output event on the transition vector in the higher-order state set, but it is optional to do so, since the reset contingency is the default. In some instructional procedures, however, it may be desirable that the lower-order procedure is deactivated but that the current state in that state set

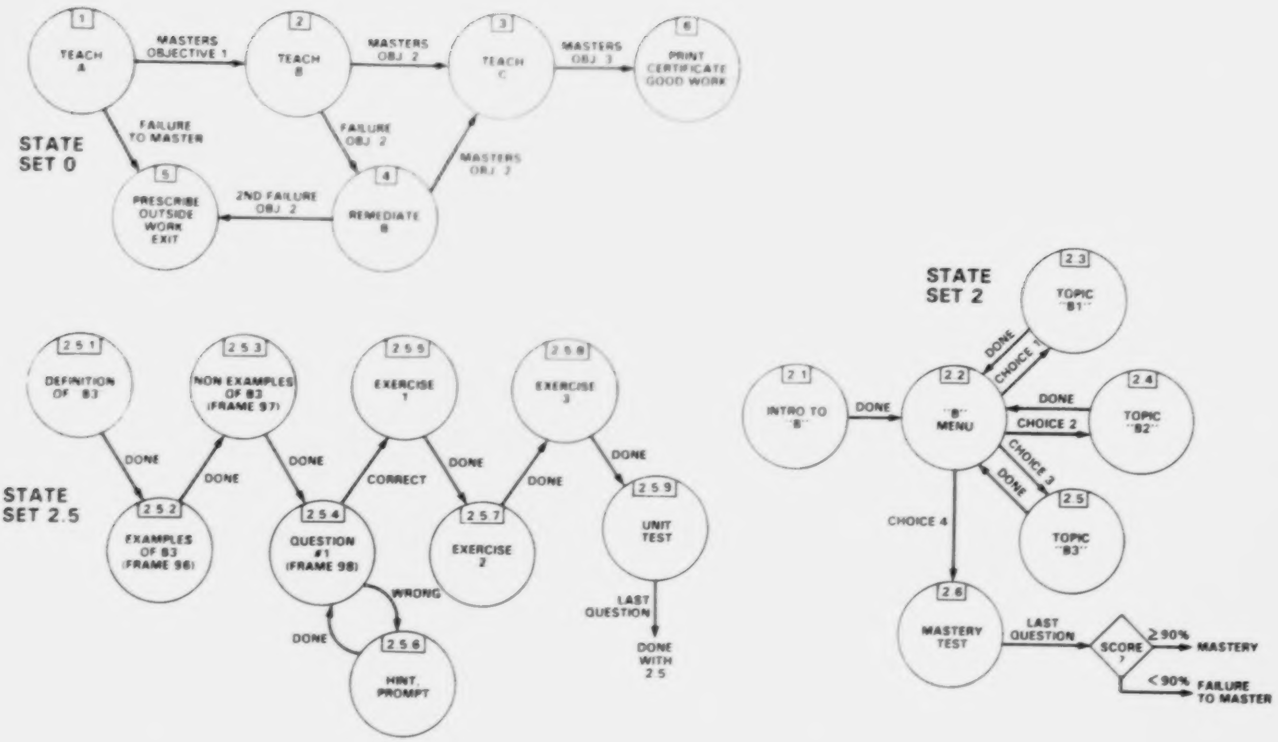

STATE SET 0
1
TEACH A
MASTERS OBJECTIVE 1
2
TEACH B
MASTERS OBJ 2
3
TEACH C
MASTERS OBJ 3
6
PRINT CERTIFICATE GOOD WORK
FAILURE TO MASTER
FAILURE OBJ 2
MASTERS OBJ 2
5
PRESCRIBE OUTSIDE WORK EXIT
2ND FAILURE OBJ 2
4
REMEDIATE B
STATE SET 2
2 3
TOPIC "B1"
DONE
CHOICE 1
2 1
INTRO TO "B"
DONE
2 2
"B" MENU
2 4
DONE
CHOICE 2
TOPIC "B2"
CHOICE 3
DONE
2 5
TOPIC "B3"
CHOICE 4
2 6
MASTERY TEST
LAST QUESTION
SCORE ?
≥90%
MASTERY
<90%
FAILURE TO MASTER
STATE SET 2.5
2 5 1
DEFINITION OF B3
DONE
2 5 2
EXAMPLES OF B3 (FRAME 96)
DONE
2 5 3
NON EXAMPLES OF B3 (FRAME 97)
DONE
2 5 4
QUESTION #1 (FRAME 98)
CORRECT
WRONG
2 5 6
HINT, PROMPT
DONE
2 5 5
EXERCISE 1
DONE
2 5 7
EXERCISE 2
DONE
2 5 8
EXERCISE 3
DONE
2 5 9
UNIT TEST
LAST QUESTION
DONE WITH 2.5

Figure 2.

is remembered so that it will be the active state the next time the state set is activated. The convention for this is to use the word SUSPEND as the output event on the transition at the higher-order state set that deactivates the state set. Figure 2 illustrates some of the conventions of state diagramming in a procedure with parallel state sets. Only a few of the lower-order state sets have been included for this example.

In practice thus far, only one lower level state set corresponds to a single higher level state, but conceptually, this higher level state can correspond to a whole family of parallel state sets in cases where complex procedures with simultaneous contingencies are involved.

One of the most critical steps in implementing computer systems based on state notation has been the definition of the order of priority and sequence of processing inputs, especially when the same event serves as input to more than one state set. In theory, input events are instantaneous, but in practice on von Neumann-type computers, there is an order of execution. It is proposed here that systems implementing the hierarchical approach to state notation will process inputs at the lowest level of the hierarchy first. Otherwise, because of the convention that lower level state sets are inactivated by state transition at a higher level, these inputs might never be properly processed. In the future, parallel processing machines may make this a moot point.

DISCUSSION

This top-down design methodology is consistent with an Instructional Systems Development, or ISD, systematic approach to instructional design (TRADOC, 1975). At the top-level state set, transitions can correspond to mastery of instructional objectives, or failure to master (GO/NO GO). At the next level down, the transitions will correspond to enabling objectives. Further down, the transitions will reflect a particular teaching strategy, such as "present rules, examples, non-examples, and then test". State notation adds a number of advantages to the ISD approach. Since it produces a family of modules, lessons are easy to debug, test, and modify. This approach lend itself to better-structured lessons. It has the advantages of outlining and flowcharting, but it handles real-time relationships better, and is probably more compatible with complex knowledge structures like semantic nets for the inclusion of artificial intelligence in CAI (Sleeman and Brown, 1983). It provides a good mechanism for communication between the designer and programmer, and allows a process of step-wise implementation with the higher-level lesson logic being tested and debugged before the lower-level frames are coded.

REFERENCES

Belady, L.A., Evangelisti, C.J., and Power, L.R. GREENPRINT: A graphic representation of structured programs. IBM Systems Journal, 1980, 19, 542-553.

Dwyer, B. A user-friendly algorithm. Communications of the ACM, 1981, 24, 556-561.

Kieras, D. and Polson, P.G. An approach to the formal analysis of user complexity. Project on User Complexity of Devices and Systems, Working Paper No. 2, 1982.

Kushner, H.K., Knapp, J.Z., and Snapper, A.G. A generalized behavioral controller. IEEE Proceedings of the 20th Annual Conference of Engineering in Medicine and Biology, Boston, Massachusetts. Washington, D.C.: MacGrego & Werner, 1967.

Mealy, G.H. A method for syntesizing sequential circuits. Bell System Technology Journal, 1955, 34, 1045-1079.

Moore, E.F. Gedanken-Experiments on sequential machines. Automata Studies, Annals of Mathematical Studies, 1956, 34, 129-153.

Sleeman, D. and Brown, J.S. Intelligent Tutoring Systems. New York: Academic Press, 1982.

Snapper, A.G., Kadden, R.M., and Inglis, G.B. State notation of behavioral procedures. Behavior Research Methods & Instrumentation, 1982, 14, 329-342.

Snapper, A.G., Stephens, K.R., and Lee, D. The SKED Software System. Manual 1: Papertape Version. Kalamazoo: SKED Users Group, 1974.

Stephens, K.R. and van Haaren, F.P. The origins and development of state notation and SKED. Behavior Research Methods & Instrumentation, 1977, 9, 72-74.

Interservice Procedures for Instructional Systems Development: Executive Summary and Model, U.S. Army TRADOC Pamphlet 350-30, 1975.

Wirth, N. Program development by stepwise refinement, Communications of the ACM, 1971, 14, 221-227.

Woods, W.A. Transition network grammars for natural language analysis. Communications of the ACM, 1970, 13, 591-606.

AN INVESTIGATION OF THE STAFF CONCERNS IN A SCHOOL DISTRICT IMPLEMENTING COMPUTER EDUCATION

Dennis D. Cole

University of Colorado

ABSTRACT

School staff development offices are faced with several issues when implementing computer technology. Background information including student experience and concerns determine how successful inservices will be. This random survey of district staff members generates a typical profile for districts beginning implementation. Results indicate staff are primarily concerned with getting information about computers and how they will effect their work habits.

INTRODUCTION

Excitement about technological innovation in education is not a new phenomenon. In fact, it is quite common (Saettler, 1968); this time the microcomputer is the source. The Corporation for Public Broadcasting reported 78% of all buildings surveyed using computers ("Here's How Schools Really Use Technology", 1984). The annual school computer use survey completed by TALMIS showed there were 730,000 personal computers in United States kindergarten to twelfth grade public and private schools. That figure doubled the 1982-83 school year figure and TALMIS projects approximately 1.2 million computers by June, 1985 ("U.S. Approaching More Than One Million Computers in Schools", 1984). Teachers, administrators, parents and students are demanding that school districts purchase computers. Some school districts have purchased based on a comprehensive computer-use plan while others are only now struggling with using their purchased computers.

Even computer educators disagree on their educational application. Arthur Luehrmann recently suggested that all students program in a computer language. In contrast, Columbia University's Mary Alice White feels that computer literacy is a passing phenomenon and suggests that schools rethink their curriculum and how they deliver instruction ("Computer Gurus Clash Over Schools' Future", 1984).

Once school districts settle on course content using computers, teachers need to demonstrate confidence using a computer and convey this confidence to students to prepare them to use computers. (Dickerson and Pritchard, 1981). David Moursund (1984) fears that a "Logo equipped computer system is not a teacher proof educational tool." (p. 3) He suggests that teachers need a substantial amount of computer education and experience to realize the potential of a Logo equipped classroom. Teachers must receive high quality preservice and inservice computer education supporting them with appropriate resources, regardless of the curriculum content.

There is difficulty agreeing on the materials and style of computer training for teachers. Gary Bitters (1983) suggests that teachers be taught computer terms, how computers work, computer operation, applications, programming, societal implications and an awareness of the direction computers are taking. He suggests that a teacher be fluent in at least one computer language such as Logo, BASIC or Pascal. Other authors include the evaluation and integration of new computer courseware into the curriculum (Rawitsch, 1981; Martin and Heller, 1982). Establishing the subject matter for computer training is difficult, but it is an important step.

There have been many attempts to address individual differences in students, but what about teachers? With many staffs receiving inservice because of recent computer purchases, have we accounted for their differences in terms of computer instruction? Teachers have seen computers in the workplace and have developed attitudes toward them. Because teachers have a variety of training, and prior to developing inservice materials for them, a district level needs assessment gathering some background on teachers at various grade levels and at different stages of using computers should be initiated (Braid, 1984). Information from this assessment should determine more appropriate offerings.

Tools exist for evaluating teacher adoption of a new technology. The Concerns-Based Adoption Model (CBAM) gives a conceptual framework for people incorporating new technology into their classroom (Loucks & Hall, 1977; Wedman & Heller, 1984; Wedman & Strathe, 1985; Hall, George, & Rutherford, 1979). Each individual reacts to change differently. These steps are generalizations of the sequence staff members take implementating an innovation. 1) A staff member's knowledge begins at zero, reflecting that the technology is unrelated to their current job requirements. 2) As the person becomes more aware, they begin to analyze the technological development's impact on them personally (i.e., how it would be useful). 3) Relating the technology to their teaching job is the next step. 4) Finally, as teachers take a more global look at the technology's impact, they begin putting it in the perspective of their assignment. This model helps staff developers to identify the concerns of the staff they are training and design instruction based on that stage of concern.

Method

This study is a collection of responses from 337 staff members in 40 schools from a large suburban middle class school district. Teachers and administrators were randomly selected from 1429 certificated employees - 681 elementary, 347 middle school, and 401 high school staff members. Approximately 42% of the respondents represented the elementary, 29% the middle school, and 29% the high school levels.

Materials

The Stages of Concern Questionnaire (SoCQ) used in CBAM (Hall and Rutherford, 1976) was revised addressing only the first four stages: Awareness, Informational, Personal, and Management. Since the majority of district schools began using computers in October, 1983, and inservice programs began in January, 1984, it was felt that the questionnaire should be shortened reflecting those four stages of concern. It was modified to specify the innovation as computers and wording was clarified to reflect district conditions. Table 1 illustrates examples of the changes which were made to all questions.

Table 1

Comparison of the CBAM States of Concern Questionnaire with the Modified Questionnaire

CBAM

35. I would like to know how this innovation is better than what we have now.

Modified Questionnaire

20. I would like to know how computers are an improvement over what I do now.

CBAM

7. I would like to know the effect of the reorganization on my professional status.

Modified Questionnaire

4. I would like to know how computers will change the way I do my job.

An attempt was made to make the modified questionnaire consistent with the first four CBAM stages. Content validity was derived from the close similarity to the SoCQ and the review of school district people working with computers or measurement. Reliability was measured by using Cronbach's formula for coefficient alpha, a generalization of the Kuder-Richardson Formula

20 for dichotomous items. Table 2 illustrates the alpha coefficients of internal consistency for the four stages measured.

Table 2

Coefficients of Internal Reliability for the Modified Questionnaire

Stage	Alphas
Awareness	.48
Informational	.62
Personal	.63
Management	.57

The questions were designed to measure the individual's affective response to computer implementation and consisted of 5 questions relating to each stage.

Concern	Definition
Awareness	Unconcerned about computers in education
Informational	Concerned about the general characteristics of computers in education and what is required to use them
Personal	Assessing present teaching style and concerned about changes necessitated by the demands of computer education
Management	Concerned about the time, organization, and management of computers in education

Respondents circled a number from 0 to 7 identifying their concerns and computer involvement with the following related meaning:

0 1 2 3
Irrelevant Not true of me now Somewhat
4 5 6 7
true of me now Very true of me now

Demographic information was collected including gender, teaching assignment, years of teaching, age, semester hours of computer related courses, strongest subject area, computer use experience and computer ownership. The combination of responses to these categories and the concerns that were expressed will provide the information needed to individualize inservice training.

Participation was excellent as 92% of the surveys were returned; elementary teachers returned 97%. Respondents may have been reacting to the fact that the school district had just spent $400,000 on computers with plans for $600,000 more to be spent over the next two years. Staff members may have used this survey to reflect their concerns about this implementation.

Procedure

The questionnaire was delivered to the principal of 375 randomly-selected staff members (elementary, middle school and high school). A cover letter accompanied the questionnaire and included a request to complete and return the questionnaire to the investigator within two weeks. Follow-up letters with a copy of the questionnaire were sent directly to the 92 non-respondents requesting them to complete the survey and return it. Fifty-four additional surveys were returned.

Design and Data Analysis

Questionnaire results were analyzed by adding the mean scores of the five questions related to each stage of concern. The totals measured the intensity of concern at the awareness, informational, personal, and management levels. High scores indicated the current stage of concern of individuals. Graphs of the scores (Figure 1) were generated for elementary (n=142), middle school (n=97), and high school (n=98) levels. A district wide profile (n=337) of teacher concerns was also created (figure 2).

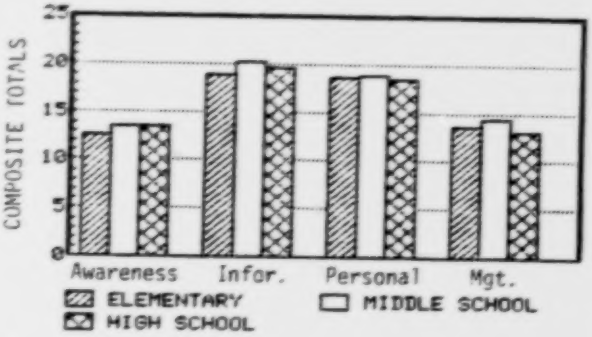


Figure 1 Computer Implementation Concerns by Level

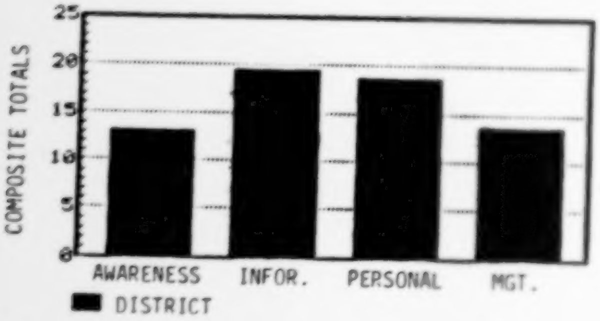


Figure 2 District Profile of Computer Implementation Concerns

Results and Discussion

The profile by levels in Figure 1 and the district profile in Figure 2 illustrate that their concerns are related to themselves (Loucks & Hall, 1977). Staff expressed intensity with regard to the informational and personal stages of concern. They reflected concern about computers and interest in learning more about them. Equally important to the training program is the high level of personal concern exhibited. Interest in how their role changes when they implement computers is a critical factor to be considered in planning inservices. Staff members are worried about how their job would change, who will control the computers they will use, and what will be required of them in terms of time and energy. Staff trainers will have to reduce these concerns in inservice courses before staff members will look at the implementation of computers objectively (Hall et al., 1979). A comfort level will need to be reached for progress to be made in additional inservice offerings and computer implementation.

The management concern is less intense suggesting that few staff have yet considered that aspect of computer implementation. The progress to management concerns would mean that orientation has changed from self-concerns to task concerns (Loucks & Hall, 1977). Personal and informational concerns are not sufficiently satisfied to facilitate movement along the continuum to management.

There are two possible interpretations of the awareness concern or lack of it. Two groups of people could have expressed low levels here. Non-users of computers could indicate by their low scores that they haven't yet had the reason to consider computers. Users who feel confident on computers would illustrate in their low scores that they accept computers and are ready to tackle other concerns. The high scores at the other self-oriented concerns would suggest that most staff members are less confident users. Since 50% of the respondents had less than one credit hour of computer training and 13% expressed that they had never used one, it would appear that most of this population make little use of computers. From conversations with administrators from other districts in the region, it is suspected that many school districts fit this profile.

Implications

Staff members surveyed in this study expressed mainly self-oriented concerns-interest in information about computers and receptivity to additional training. Instruction should concentrate on the initial level of computer use and give special attention to staff members' personal concerns such as changes in job habits resulting from computers and the individual time/energy that will be required of them.

Few of the district staff are ready to consider issues related to computer based instruction. Inservices incorporating information such as computer managed instruction and word processing are appropriate to only a few. Attempts to incorporate more complex uses of classroom computer instruction at this time will likely fail. The school district should plan additional instruction for all staff to meet the self-oriented concerns and prepare teachers to reach out for different computer uses. Further examination of the survey information should result in profiles for each level and academic discipline which will facilitate more individualized instruction in staff development courses.

REFERENCES

Bitters, G.G. (1983). Computer literacy for teacher certification. Educational Computer Magazine, 3, 22.

Baird, W. E. (1984). A road atlas for computer literacy and teacher training. The Computing Teacher, 12(3), 11-16.

Computer gurus clash over school's future. (1984, July 23). *Education USA*, p. 358.

Dickerson, L. and Pritchard, W. H. (1981). Microcomputers and education planning for the coming revolution in the classroom. *Educational Technology*, *21*, 7-12.

Hall, G. E., George, A. A., & Rutherford, W. L. (1979). *Measuring stages of concern about the innovation: A manual for use of the Soc questionnaire*. Austin, TX: The University of Texas.

Hall, G. E. & Rutherford, W. L. (1976). *Measuring stages of concern about the innovation: A manual for use of the stages of concern questionnaire*. Austin, TX: Research and Development Center for Teacher Education, The University of Texas.

Here's how schools really use technology. (1984). *School Tech News*, *2*(1), p. 1.

Loucks, S. F. & Hall, G. E. (1977). Assessing and facilitating the implementation of innovations: a new approach. *Educational Technology*, *17*(2), 18-21.

Martin, C. D. and Heller, R. S. (1982). Computer literacy for teachers. *Educational Leadership*, *39*, 46-47.

Moursund, D. (1984). Logo frightens me. *The Computing Teacher*, *11*(5), 3-4.

Rawitsch, D. G. (1981). Teaching educators about computing: A different ball game. *The Computer Teacher*, *9*.

Saettler, P. (1968). *A History of instructional technology*, New York: McGraw Hill.

U. S. approaching more than one million computers in schools. (1984, November). *The Computing Teacher*, *12*(3), p. 9.

Wedman, J. & Heller, M. (1984). Concerns of teachers about educational computing. *Association for Educational Data Systems Journal*, *17*, 31-40.

Wedman, J. & Strathe, M. (1985). Faculty develoment in technology: A model for higher education. *Educational Technology*, *25*(2), 15-19.

REDUCING LEARNING ANXIETIES WITH COMPUTER-BASED INSTRUCTION

Albert L. Harris - Clayton Junior College

Jacqueline M. Harris - North Springs High School

ABSTRACT

Learning anxieties pose a major obstacle to education and training. In this paper, eight causes of learning anxieties are identified and discussed. The paper then centers on ways in which computer-based instruction (CBI) can reduce learning anxieties. Each cause is reviewed individually. Next, three problems that arise when using CBI to reduce learning anxieties are addressed. Finally, some conclusions are presented for further thought.

INTRODUCTION

Computer-based instruction (CBI) is coming into its own as a educational tool at the corporate level and in educational systems. CBI is not a new tool or technology, however, we are now learning how CBI can be most effectively used, especially in reducing learning anxieties. Computer-based instruction is nothing more than a method of instruction in which a learner and a computer interact with one another. This two-way communication produces human learning and retention of this learning. CBI takes the name of computer assisted learning (CAL), computer assisted instruction (CAI), or computer managed instruction (CMI). For our purposes, these are all forms of computer-based instruction.

Learning anxieties pose a major obstacle to education and training. Learning anxiety is a fear or an intense, negative emotional reaction to learning. It may be isolated to one subject, say mathematics, or it may be a general phobia against any learning experience. It is perhaps more publicized as a problem in educational systems (colleges and universities, high school, and even at the grade school level) but it also hinders training at the corporate level and increases costs and the overall effectiveness of all training. There are many levels of learning anxieties, from mild to overpowering. There are also many causes.

CAUSES OF LEARNING ANXIETIES

Eight (8) major causes of learning anxieties have been identified. To simplify discussion, they can be placed in three broad categories: student-related; instructor-related; and instruction-related. Exhibit 1 presents the categories and causes of learning anxieties. A brief discussion of each cause is presented in the following paragraphs.

Student-related

Student-related causes of learning anxieties include lack of confidence and negative student attitudes. While these causes can be influenced by the instructor and teaching methods, they are more a function of the individual student's background and mental attitude. Probably the strongest cause of learning anxiety is students' lack of confidence in their own ability to learn. The fear of working on one's own and making mistakes is also very strong. Students are reluctant to participate in an activity in which they feel they will have little chance of

EXHIBIT 1

Causes of Learning Anxieties

Student-related

1. Lack of Confidence
2. Negative Student Attitude

Instructor-related

3. Instructor Bias
4. Authoritarian Teaching

Instruction-related

5. Lack of Variety
6. Lack of Relationship With the Real World
7. Emphasis on Memorization
8. Emphasis on Speed

success. Further, the requirement that the student do his/her own work places another strain on their confidence. In most learning situations, students must do their own work and assimilate the information to effectively learn the material.

A negative student attitude is both a result and a cause of learning anxiety. People tend to dislike that which they fear or do not understand. The result is often a negative attitude toward learning. Often this attitude is reinforced by others, who also may have a negative attitude toward the subject. "Learning is not supposed to be fun" is a common theme among students. This theme highlights the negative student attitude cause.

Instructor-related

Instructor bias and authoritarian teaching are the two instructor-related causes of learning anxiety. "All of us are subject to inherent prejudices, often at the subconscious level" (Bork, 1980, p. 64). These unconscious personal feelings affect the teaching and the students' receptiveness to learning. Hard as an instructor tries to hide them, his or her unconscious biases cause learning anxiety. "Teachers treat male and female learners differently" (Fennema, 1980, p. 172). "Minorities are often treated as though they lack ability" (Kennedy, 1984, p. 68). Many times different levels of classes are held, some for the 'smarter' students and some for the 'slower' students. The instructor knows the difference and so do the students.

Authoritarian teaching occurs when instructors imply that their way is the only way to solve a particular problem. Learning anxiety may be produced in the students when the instruction tends to be a lecture or information giving session. This authoritarian approach is not necessarily intentional by the instructor. Instructors spend many hours trying various approaches to explain, derive, or show a particular concept and inevitably a student will sum up the efforts, "Well, if you say its so then I will just memorize it." There are often too many levels of intuitive understanding or lack of understanding for a instructor to explain or demonstrate an idea satisfactorily to each student.

Instruction-related

The way an instructor teaches has a large impact on learning anxiety in students. The four causes in this area include lack of variety, lack of relationship with the real world, emphasis on memorization, and emphasis on speed. A lack of variety occurs when the instructor becomes so familiar with the subject that the teaching becomes staid.

Students often complain that lessons have no relationship to the real world and have no (or little) application or interest outside the classroom. Much of the instruction seems segmented into learning chunks to be consumed by the student with no overall interrelation between the chunks. This causes a lack of interest and anxiety toward learning.

Memorizing facts or procedures for quick recall is one method of learning. We start in grade school memorizing facts and continue throughout life. Emphasis on memorization thus becomes a cause of learning anxiety. Emphasis on speed is also a cause. Even good students become flustered when the time allotted for a lesson or test seems insufficient. It is difficult to pace instruction so the better students will not become board and the slower students will not become overwhelmed.

USING COMPUTER-BASED INSTRUCTION

CBI is a proven tool in reducing learning anxieties in students of all ages. To see how it is accomplished, we shall look at each one of the causes of learning anxieties.

Lack of Confidence

There are a number of ways in which CBI can help students gain self-confidence. Computers ". . . (a) have infinite patience, (b) never get tired, (c) never get frustrated or angry, (d) never forget to correct or praise and (e) individualize learning" (Lawton and Gerschner, 1982, p. 51). Students, thus, have a positive attitude about using the computer to learn. They are not embarrassed by mistakes they make on the computer, since only the student and the computer need be aware of the mistake. Feedback given by the computer on errors made by the student are immediate, relevant, direct and diagnostic. CBI allows the student to correct himself/herself and answer subsequent questions correctly, thus increasing confidence. The feedback is not judgmental or punitive.

CBI can now be designed so that the student cannot lose; he or she is always successful to some degree. This success allows the student to gain more understanding of the ideas or principles to be mastered, become more skillful and efficient, and increase confidence to learn the subject. The skill level at which a student may start with a particular CBI program may be chosen by the student, the instructor, or the CBI itself, after a few initial problems by the student. The more progressive CBI programs are designed to give students problems at a level where the student will be challenged, but he or she will be successful the majority of the time.

Successful experiences, positive re-enforcement, and the proper skill level will bolster the student's self-confidence and help to overcome some of the student's learning anxieties.

Negative Student Attitudes

The computer itself has the potential to be the means to bypass the negative attitude mental block. The computer has a fascination for students of all ages; it can be a fun challenge. Students do not classify working with a computer as learning and *yet they* learn. Since the computer is associated with a myriad of applications, including games, it is easy to get students to try CBI. If the programs are wisely chosen, the student will gain an understanding of the concepts being taught before they realize that they are learning.

Kulik's (1984) research into attitudes about computer learning found (1) students' attitudes toward the subject are more positive where CBI is used than where it is not, (2) students' attitudes towards computers are more positive when CBI is used, and (3) students that use CBI show a slightly more positive attitude towards instruction than students who do not use CBI. There is evidence that computers can affect student attitudes and behaviors in positive ways (Fisher, 1984)

Instructor Bias

The computer has no prejudices. It cannot perceive the physical or socio-economic traits of the student that uses the CBI. Previous grades cannot subjectively influence the computer. Part of the success of CBI is that the computer has a lack of subjective evaluations. The computer bases its evaluations strictly on student performance, not on the personal characteristics of a student or his or her social relationship with the instructor (Clement, 1981). One word of caution, CBI will never replace the instructor. The personal contact with the instructor is a vital part of the educational process.

Authoritarian Teaching

Some programs provide a means for a student to examine his or her intuitive ideas about a subject and manipulate these ideas into a more complete and accurate understanding of the subject. LOGO, for example, allows a student to manipulate information or ideas via the computer until the student intuitively feels that the concept makes sense and accepts it. Student exploration of ideas through student controlled computing, computer simulation, or other forms of CBI can lead the student to a more complete understanding and integration of ideas into his own thinking. The variety of means to approach an idea via the computer or the different ways a computer can allow an idea or concept to be examined can lessen student learning anxiety, since it can present more than one correct way to learn any given concept or idea.

Lack of Variety

Although an instructor with imagination and enthusiasm can put variety into the instruction, the use of CBI can offer yet more variety to students. CBI accomplishes this in three ways: (1) there is a variety of CBI games, tutorials, and simulations that may be used to enhance learning; (2) there are many varied levels of ability for the students, including slow, average and advanced; and (3) use of computers frees the instructor to devote more time to personal, human considerations of students.

Studies have shown that instructors with computers spend more time working individually with students, than instructors without computers. This one on one relationship itself adds infinite variety to the learning situation.

Lack of Relationship with Real World

The relationship of the subject to the real world is important to reduce learning anxiety. It is easier to learn a particular fact or idea if it can be related to the real world. CBI can create situations which are not available in convenient form for the students to play with and explore the possibilities. It can make use of real world simulations, imaginary simulations or game situations to demonstrate interaction with our everyday lives.

As the quality of CBI increases, we will find programs that unite several disciplines in a real world situation. Taking a subject out of its isolation and uniting it with other subjects will demonstrate to students the vital role the idea plays in our real world.

Emphasis on Memorization

CBI can help the student retain facts and procedures through repeated exercises and manipulation of ideas with the computer. These are the first steps to help students retain important facts or procedures. Next, the student must practice or use the facts or procedure. The computer and CBI has already been used for skill and drill practice in classrooms. With more sophisticated computer programs, this use can be extended beyond the textbook variety of problems or examples. CBI can give an endless variety of problems relating the new ideas to other facts or procedures already learned. It can ask the same questions several different times and several different ways. It can tailor the problems to fit the level of ability of the student and change this level of difficulty as the student progresses. These capabilities of CBI can assist the student to memorize facts or procedures in an atmosphere devoid of many of the learning anxiety producing factors.

Emphasis on Speed

CBI can assist the instructor in reducing learning anxiety due to the need for speed. Speed is de-emphasized when the student uses CBI. Even the same student on different days or with different subjects may assimilate ideas at varying speeds. Robin (1982) noted that on Mondays students chose to do fewer mathematics problems than on other days of the week. An instructor cannot vary the pace of the instruction for each individual student. A computer will allow students to work at their own pace. Many CBI programs are designed to encourage students to increase their speed as well as accuracy, to beat the game or get a better score on the skill builder (Piestrup, 1984). By encouraging students to work to the limits of their capabilities, but not penalizing them for being slower or more deliberate than other students, the computer and CBI can lessen learning anxiety.

PROBLEMS WITH USING CBI

CBI can reduce learning anxieties, but the help is not without problems. Three problems are addressed; there may be more.

Computerphobia

Use of CBI to overcome learning anxieties will obviously not succeed if the student or instructor is afraid of the computer. Computerphobia is a very real problem for instructors and students alike as more computers move into the learning environment and more instruction is consigned to computers. Computerphobia is more often a problem for the instructor, not the student. If an instructor has computerphobia, he or she may choose not to have or use computers. If computers are used, a computerphobic instructor will limit the computers effectiveness by transmitting this reluctance to use computers to the students, by poor choice in the type of CBI, or by ineffective use of CBI (Jay, 1981).

Software and Hardware

CBI is only as good as the quality of the programs that are available. CBI has improved over the past twenty years and now is in a dramatic stage of flux. The first CBI programs produced were textbook-like skill and drill or tutorials. Next came computer games with an overlay of facts of limited educational value. Now CBI includes simulations of real and imaginary worlds, game builders and student directed exploratory learning. Game builders are different from the earlier learning games in that the concepts are inherent to the game, not just a later addition. As the students master the concepts and ideas the next level of play builds on these ideas. More understanding leads to more sucessful play. This type of CBI was developed through combined efforts of educators and computer experts and should result in a higher quality of CBI.

Cost of software and hardware is a primary consideration in the use of CBI. Computers can reduce learning anxiety only if they are accessible. This means computers or computer terminals must be available and judiciously placed for maximum use. The more the computer can be used by students, the less the cost of the machine and software per student hour of use.

Changes in Instruction Methodology

CBI has opened up new approaches to instruction and learning. To make full use of these innovations, changes in instruction methodology will be necessary. Even if we start slowly in implementing the use of CBI in training sessions or classroom situations, with some time alloted for using the computers for explorations or skill and drill, instruction methodology changes will be necessary. Homework can be assigned on an individual basis rather than a class basis. Instructors can take the various levels of homework completion into consideration when they plan their next lecture or session. Testing may also change in keeping with the more individualized assignments.

CONCLUSION

CBI can be a tremendous help in training and education, especially with the ability to help students who suffer from learning anxieties. Computer-based instruction provides the link between the student and the machine. The use of CBI is an emerging tool for increasing the learning capabilities of all students, especially those with learning anxieties.

There are, however, obstacles to overcome. The hardware and software necessary can be expensive. Quality CBI software is just now becoming available thanks to the merging of educators' and computer scientists' efforts. Finally, the individual instructor must recognize the value of CBI in learning, must acquire the necessary training, must have the hardware and software readily available, and must incorporate these tools into the instructional methodology and learning environment.

REFERENCES

Bork, Alfred. Interactive Learning. In Robert Taylor (Ed.), *The Computer in the School: Tutor, Tool, Tutee.* New York and London: Teachers College Press, 1980.

Clement, Frank J. Affective Considerations in Computer-Based Education. *Educational Technology*, April, 1981, pp. 28-32

Fennema, Elizabeth. Teachers and Sex Bias in Mathematics. *Mathematics Teacher*, March, 1980, pp. 169-170

Fisher, Glenn. The Social Effects of Computers in Education. *Electronic Learning*, March, 1984, pp. 26-28

Jay, Timothy B. Computerphobia: What To Do About It. *Educational Technology,* January, 1981, pp. 47-48.

Kennedy, Leonard M. *Guiding Children's Learning of Mathematics* (4th ed.). California: Wadsworth Publishing Company, 1984.

Kulik, James A., Bangert, Robert L., and Williams, George W. Effects of Computer-Based Teaching on Secondary School Students. In Decker F. Walker and Robert D. Hess (Eds.), *Instructional Software: Principles and Perspectives For Design and Use*, California: Wadsworth Publishing Company, 1984

Lawton, Johnny and Gerschner, Vera T. A Review on the Literature on Attitudes Towards Computers and Computerized Instruction. *Journal of Research and Development In Education*, 1982, *16*, 50-55

Piestrup, Ann. Game Sets and Builders. *Byte*, June, 1984, *9*(6), pp. 215-218

Robin, Lisa. Wild About Math. *Media and Methods*, March, 1982, pp. 6-7

1000 REQUESTS/WEEK ON LIMITED RESOURCES

Joan Lauer Hayes

Western Washington University

During 1984 - '85, three microlabs maintained and staffed by the Western Washington University Computer Center served different purposes. One lab gave priority to Education Department users, and one to Journalism students. The third was a general-purpose lab with use so high as to require a disproportionate amount of supervision and concern. Relative uses of the three labs is displayed in Figure 1. (One of the recommendations accepted by the Computer Center, but not always enforceable, is that a maximum of 30% of the hours in any lab be used for classroom instruction.)

The main microlab complex was made up of two rooms: one containing 20 IBM PC's with 512K memory, and one containing 10 IBM PC's with 256K memory. These adjoined a central facility in which were 30 terminals accessing VAX and IBM mainframes. Staffing was provided 60 hours a week until the end of the quarter, when additional hours were added.

Users of the facility were students in BASIC, COBOL, or Cartography classes accessing the mainframes, or were microcomputer users learning Lotus 123, Symphony, Microdynamo, BSTAT, Wordstar, DBaseII, LOGO, or IBM PILOT.

Because the nature of consulting questions was diverse, the facility became a training ground for Computer Science students who had not taken enough advanced classes to work in the main terminal room. In addition, students who were relatively new to microcomputing, but who were interested, had good skills in working with people (and, hopefully, had Work Study funding!), were invaluable. Two categories of staffing were in effect: Consultant and Aide. Aides who learned quickly functioned at a Consultant level at the end of the year. The Aide category was created partially to protect staff members who were not able to answer programming-level questions. During each hour, there were two people on duty: 1) an Aide (usually) who stayed at the desk for security and service reasons, and distributed software and, 2) one roving staff member (usually a Consultant) who was able to answer questions at terminals or micros. All staff members were asked to record each time they were asked for service or information. Throughout the quarter, the number of requests doubled those in the North Campus terminal room, where 55 mainframe users could be helped by a single Consultant.

To meet the needs of novice microcomputer users, Familiarization Sessions covering both hardware instructions and lab procedures were given daily (outline available on request) and required for all new users. Faculty members who wished to give their own class Familiarization Sessions were offered the services of a Computer Center student staff member to assist students who were having keyboard problems, and to explain the lab procedures. A reservation system was established for classes using the lab, and standards written up for class use.

The sorts of problems encountered by the staff in assigning users to micros included the following:

> Symphony users required the 512K machines.
>
> Students currently in a class requiring Symphony had priority over users no longer in a Symphony class.
>
> Word processing users had to be on a class list submitted by a faculty member who was either teaching or requiring word processing. (Otherwise, the lab would have been impacted by word processing students.)
>
> Only certain machines (fewer than half) could be reserved.
>
> Users sometimes moved from one micro to another for reasons unrelated to the system!

Several times each quarter, it was necessary to place a one-hour limit on users, or at least to ask the person who had arrived earliest to be the one to give up a machine.

USER MONITORING SYSTEM (UMS)

Many problems related to machine assignment were resolved by the authoring (by Head Consultant Matt Henson) and modifying (by Head Consultant Mary Graham) of a database program written in Pascal which runs on the South Campus VAX 11/780.

Before a student is assigned a PC, a staff member checks the student's status on UMS. If the name of the potential user has not been entered, it's probably necessary to attend a Familiarization Session or to otherwise have competency in micro use established. If the potential user has attended a Familiarization Session, but wants to use Wordstar and is no longer authorized for Wordstar use, the staff member can suggest courses of action.

It is possible to search the file by name or student number; on request, persons with the same last name are listed with first names and student numbers. Time of arrival is recorded with the assignment of a machine. "Order of Arrival" lists can be displayed whenever necessary. And it is easy to deal with a move from one machine to another; this is particularly important when users are asked to empty a lab (for class use) and want to move to the other lab.

Because individual reservations are not always filled, staff members have chosen to eliminate the reservation system from UMS. But it is possible to display a picture of either of the two labs, and see which machines are still available before making assignments.

UMS has made a positive impact on both staff and users. The program is perceived as being systematic, so that people who are waiting for a micro sense that they will be treated fairly. The problem is then seen as being a matter of limited resources rather than a personal issue. From a staff person's point of view, the only problem is in being asked to take the time to run each user through the program when there are long lines of people waiting for service.

MACHINE ASSIGNMENT AND SOFTWARE DISTRIBUTION

After trying several methods, last year's staff decided on an inflexible request for student ID, a driver's license, or a student meal card when users were signing out software and/or documentation.

Each piece of software has a "library card pocket" attached to it, with a card identifying the piece of software and its number.

Each micro has been assigned a number, and behind the staff desk there is a rack of cards for each room.

When a student arrives, the UMS entry is checked, a machine is assigned, the card for the micro is removed from the rack, the cards from the software and manual are removed from their holders, and all are put in an envelope with the student's identification card.

When the student leaves (and often many leave at the same time), the staff member is asked to check people out one at a time, and to make sure that all ID and cards get back into the right slots. If staff members are not assertive enough about this process, it's easy to give students the wrong ID.

COPYRIGHT PROTECTION

The Washington State WAC 516 - 22 - 020 Code and Chapter 9A.52 RCW make clear our regulations against copying software. It is posted.

We have been consistent in not providing DOS disks for students, and handle formatting and diskcopies if they are needed, sometimes on a class basis.

If users show up with their own Wordstar disks (identifiable by a different set of colors), we do not accept that software and talk with them.

If users make use of copying programs, we ask them to leave.

GAMES

No games, unless assigned by a faculty member.

SUPERVISION AND TRAINING

As the number of labs and student staff members has increased, more operational responsibilities have shifted to Head Consultants in each lab. The Head Consultants meet three times a week with the Coordinator, and consult with her in emergencies.

Student staff members are expected to attend a weekly meeting, where procedures are discussed and tutorials presented. In addition, attendance is expected at several Saturday workshops. Two of the workshops were held with student staff members from the Tutorial Center, who deal with similar problems with users. The workshop topics dealt with Assertiveness (Fall workshop) and Burnout (Winter Quarter). And there are parties -- important but not required.

Over the years, a 70-page manual has evolved, containing most of the information we can predict will be needed by consultants and aides. In order to make sure such a document will be read, there are short quizzes at meetings.

Ten hours/quarter is provided for training on new packages or systems. The staff has pointed out that the most effective way for them to make use of that option is to require hard copy demonstrating knowledge of the program by a particular meeting.

We have been aware that we demand a great deal of Computer Center student staff. There were about 35 students working for the Academic Services area of the Computer Center last year. We are fortunate in that the training and experience are valuable to the students, and that they continue to be eager to work for us.

In 1985 - '86, it is expected that the number of microlabs will double (to a total of 6, containing close to 100 machines). The cost of staffing and operating the increased labs has been referred to as "The $64,000 Question". If the $64,000 is not available, training of novices and volunteers provided by academic departments may be the only means of staffing. Our dependence on existing systems will increase, and the need for others will become apparent. The most obvious need will be for a microlab scheduling program, currently under development.

NOTE: At this writing, use of the primary microlab has increased over 50% to almost 1700 requests per week. Although the frustration level is high for students needing to use the machines, lab operations are still considered to be smooth, and the student staff is working well together. We feel fortunate in having been able to develop systems and to train staff prior to demands placed on them by excessively large numbers of users.

Figure 1 MICROLAB USE SPRING QUARTER, 1985: WEEKS 1-9

LAB	TOTAL HOURS OPEN	HOURS FOR RESERVED WWU CLASSES*	HOURS FOR RESERVED CONT. ED., OTHER CLASSES**	TOTAL DISK SIGNOUTS	TOTAL CONSULTATIONS
AH 01, AH 02 MAIN MICROLAB	558	340 61%	36 6%	5,301	3,534
CH 106 JOURNALISM LAB	558	132 24%	0	824	1,008
MH 350 MH 350A EDUCATION LAB	594	340 57%	20	3,525	Not Available
TOTAL	1,710	812	38	9,650	>4,542

1,405 users (17% of total student population) are authorized to use the Arntzen Hall microlab. 403 of these are in classes using WordStar, and 260 in classes using Symphony.

NOTE: Non-class Symphony users have a priority lower than class members. Only those persons authorized to use WordStar for class are permitted to do word processing in AH for projects unrelated to other programs such as Lotus 1-2-3.

* We expect each hour of class lab use to generate an additional three hours lab use for completing assignments.

** We expect each hour of Continuing Ed. class lab use to generate one additional hour of lab use.

A PLAN FOR COMPUTER LITERACY

Jan Powell

Francis Tuttle Vo-Tech Center

ABSTRACT

Computer Literacy for both secondary and post-secondary students at the Francis Tuttle Vo-Tech Center became a reality during the 1984 school year. Three levels of literacy were defined: 1) Computer Awareness; 2) Application Awareness; and 3) Programming Awareness. Levels one and two are on-line and level three is in the pilot stage. Software is used in all three levels to aid hands-on learning utilizing 15 IBM Personal Computers in the Learning Center.

INTRODUCTION

The Francis Tuttle Vo-Tech Center is committed to an open-entry, individualized, performance-based instructional program which provides quality training in 27 vocational and technical programs. This instructional methodology allows individuals to begin at various entry points and progress through a course at their own pace and ability. Experts believe job skills go far beyond the development of technical skills expertise. Each student at Francis Tuttle Vo-Tech Center receives help to develop other skills such as oral and written communications that will contribute to a successful work life.

Administrators at the Center believe that every student enrolled in a vocational program should have exposure to computer technology through the use of microcomputers. Thus, a Computer Literacy plan was formulated in the Fall of 1983.

DEFINITION

Computer literacy definitions vary from institution to institution. At Francis Tuttle Vo-Tech Center, computer literacy is defined as a basic working knowledge of how a computer functions and its influence on individuals. It includes awareness of computer applications in areas that affect industry, as well as an awareness of the skills necessary to make the computer a viable tool now and in the future.

DEVELOPMENT

A committee consisting of the Learning Center coordinator, microcomputer curriculum specialist, data processing instructors and curriculum staff members met to develop the guidelines for creating the Center's computer literacy curriculum. The group was charged with the task of developing a program which would be appropriate for delivery to every student in the school, including special needs students.

After reviewing various computer literacy programs from across the country, the committee decided that none met the exact needs of Francis Tuttle Vo-Tech's student population. A decision was made to develop a specialized program tailored to meet our needs.

Development plans centered around three levels within the computer literacy curriculum. Level I was Computer Awareness, Level II was application Awareness and Level III was Programming Awareness.

Each level would build on the knowledge learned at the previous level. All units of instruction would be individually paced using Learning Activity Packets (LAPs) so students of various academic backgrounds and learning styles could work at their own pace to master the Literacy LAPs.

Each LAP in the Computer Literacy series includes performance objectives with tasks to accomplish, learning activities with assignment sheets and specific references to a videotape and/or software application. Every student receives a LAP to use as a learning guide.

DELIVERY

Delivery of the program was planned for the school's Learning Center, because there were 15 IBM Personal Computers available in a single location. Each IBM PC has 256K with two 360K diskette drives, so basically any software package could be used. There are also five beta videotape decks that can be scheduled for viewing videotapes.

Instructors schedule both secondary and post-secondary students to take each level of the Computer Literacy curriculum through the Learning Center and receive feedback on student performance through testing vehicles. An exam is given at the conclusion of each of the three levels of Computer Literacy.

The major goals of Francis Tuttle Vo-Tech's Computer Literacy program were to help the students feel comfortable using a keyboard, disk drives, a monitor and floppy disks, as well as an awareness of how computers can be used at home and work. Students will not become programmers or highly-experienced word processors after taking all three levels of Computer Literacy.

LEVEL I

Computer Awareness is approximately four hours of instruction consisting of four Learning Activity Packets which utilize commercially-prepared videotapes and computer software. Two videotapes are used, The Computer: What It Can - and Can't Do (The Center for Humanities, Inc., 1984) and Personal Computing Concepts (SRA, Inc., 1983). The software utilized includes an Educational Demonstration (IBM), Exploring the IBM Personal Computer (Digital Learning System, Inc., 1983) and PC Introductory Set (FriendlySoft, Inc., 1983). This phase was piloted in January 1984. All students except those in Data Processing completed Literacy Level I. The goals of the Computer Literacy program are covered throughout the year within the Data Processing curriculum, so instructors thought Computer Literacy would be repetitious.

The major goals of Level I were for a student to describe the uses and limitations of computers, to demonstrate a basic understanding of the parts of a computer by using keyboarding skills in conjunction with software packages, and to define elementary data processing concepts.

LEVEL II

Application Awareness is three to five hours of instruction provided through three Learning Activity Packets which utilize both a book, A 60-Minute Guide to Microcomputers (Hollerbach, 1982) and a commercially-prepared software package, PC Master (Courseware, Inc., 1983).

The major goals of Level II were to give a student an overview of what is software, to distinguish between system software and application software, and to provide an overview of commonly-used application software which includes a word processing application, spreadsheet application and a database application.

The targeted audience is all students, with the exception of Data Processing and special needs students. This segment emphasizes basic information about software and is designed to provide an overview of commercially-prepared software packages. Level II became operational in the Fall of 1984.

LEVEL III

Programming Awareness consists of three to five hours of instruction through the usage of three LAPs. This level is in the developmental stage, with pilot testing scheduled for November, and is to be used by students beginning in January 1986. The targeted population is all students, except for Data Processing and special needs students.

The major goals of Level III are to provide an overview of the various purposes and functions of programming languages and to introduce a programming language--BASIC--whereby students learn simple commands and program logic through the creation of a simple BASIC program. An interactive, commercially-prepared software package, Fundamentals of Programming the IBM Personal Computer, An Introduction to BASIC (SRA, Inc., 1984), will be used as the primary delivery source.

EVALUATION

After completion of each literacy level, a student evaluation form is completed by each student and returned to the Learning Center staff. The form is then sent to the microcomputer curriculum specialist. The results are compiled and problem areas in the program are modified according to the student evaluation reports. Areas which receive negative comments are reviewed and new media materials are provided for replacement. Through this documentation, minor revisions are being planned in Literacy Level I. Literacy Level II has received positive reviews from the students due to the software package utilized.

CONCLUSION

Actual usage of the IBM Personal Computer by over 700 students in a given school year has created students who are no longer afraid of microcomputers. These students feel confident that they can operate an IBM PC. The staff believes that the Computer Literacy curriculum at Francis Tuttle Vo-Tech Center meets the needs of a very diversified student body. Students have become more knowledgeable of the way in which computers affect them in society through the hands-on experience gained at the Center.

REFERENCES

Bell, Trudy E., "Computer Literacy: The Fourth 'R'," Personal Computing (May, 1983), pp. 63-69.

Klassen, Daniel, "Computer Literacy: What It Is and How To Get It," School Business Affairs (May, 1983), pp. 44-45.

Powell, Arthur B. and Varden, Stuart A., "Computing Literacy for Working Class Adults," Interface (Winter 1983/84), pp. 48-52.

RESOURCES

The Computer: What It Can - and and Can't Do. New York: The Center for Humanities, Inc., 1984. (Videotape/slidetape)

Educational Demonstration, International Business Machines.

Exploring the IBM Personal Computer. Digital Learning Systems, Inc., 1983. (Diskette)

Personal Computing Concepts. Arizona: Science Research Associates, 1983. (Videotape)

PC Introductory Set. Texas: FriendlySoft, Inc., 1983. (Diskette)

Hollerbach, Lew A. A 60-Minute Guide to Microcomputers. New Jersey: Prentice-Hall, Inc., 1983. (Diskette)

PC Master. Courseware, Inc., 1983. (Diskette)

Fundamentals of Programming the IBM Personal Computer. Science Research Associates, Inc., 1984. (Diskettes)

Pilot Project: LOGO for Preschoolers as an Aid for Teacher Preparation in Technology

Dr. Charlotte Scherer

Bowling Green State University

ABSTRACT

Preparation for use of technology by preservice and inservice teachers needs to be carefully planned and implemented. Pilot projects during "downtime" can ensure that later full implementation with college faculty and students is smoother and more successful. Two pilots recently completed at Bowling Green State University using LOGO with preschoolers are described and the results summarized.

With the growing need to address the advent of technology in teacher education at both the preservice and inservice levels, it is clear that projects for involving teachers and college of education faculty in such preparation need to be carefully thought through and even piloted in advance before they become part of the regular program. One way to accomplish this is to utilize "downtime" in the academic year, and to make use of university facilities and some personnel to try out some innovative technological programs with community schools before trying to incorporate them into the established preservice program. This plan has several advantages: 1)inservice personnel are involved during the pilot, 2)faculty and staff become comfortable with what works and doesn't work in this relatively new field of trial and error, 3)local school children benefit, and 4)concrete plans can then be made for a program which continues during the academic year, and which involves preservice as well as inservice teachers.

The pilot project that was designed and implemented by the Director of the Clinical Experience Lab at Bowling Green State University involved training local nursery school teachers in the use of the computer and in LOGO, a programming language suitable for use with preschoolers and early childhood grades and then having the preschoolers come into the Lab during the break between spring semester and the first summer session to learn to use LOGO and other preschool computer resources. The teachers, who were interested in computers, but generally apprehensive and not at all "literate" with respect to computers, came in for two sessions of two hours each. In these sessions they had "hands on" experience and were introduced to the computer and to LOGO as well as two or three other preschool software packages. Following this, it was arranged to have 18 five year olds come in groups of nine to the Lab to learn about computers and learn to use LOGO to produce designs and pictures that can be printed out. There were also two four year olds, the sons of two of the Lab staff that helped with the children. This provided the Director, who has been trained in the past in working with young children and who is proficient in LOGO, with a good group to "try out" some ideas concerning how to work with young children to introduce them to computers and to the LOGO Turtle environment.

It was also important that detailed lesson plans were produced for each of the sessions. These plans helped those assisting to know the objectives for each session and how best to help. They also were a ready reference for future planning and ensured that a logical sequence of events was followed for the children's learning and skill development. Since the plans were written on the computer, they could easily be revised during the project implementation time and at any future time when being used for planning similar projects.

The children came in for 45 minute sessions three mornings a week for four weeks for a total of 11 sessions (due to the Memorial Day holiday). It turned out that several of the children were exposed to computers at school, and others had a computer at home; but some had never used one and none was familiar with LOGO. LOGO was chosen because of its reputation for putting children in command of the computer while other programs tend to make the computer in charge of the child. It was also chosen because it purports to help children think methodically and use problem-solving skills. LOGO's color turtle graphics readily lend themselves to use by young children who can make designs, shapes, and "pictures" using the single stroke version that can either be created or comes ready made with at least one version of LOGO (Abelson,1982; Papert, 1980; Rieber,1985; Watt,1982).

It is important for very young children to have many concrete experiences, including hands on experiences first, then move from these concrete experiences to a somewhat abstract level, and finally to move to the much more abstract level of a computer screen. Hence, many activities and games were planned which had the children moving like turtles with their bodies, then moving turtle objects, next programming a Big Trak toy tank (Milton Bradley, 1979) and a Turtle Tot floor turtle (Harvard Associates, Inc., 1983,1984) and finally working on the computer itself with the screen triangle that is the LOGO turtle. Following such a sequence allows for a great deal of initial and immediate success, with little or no frustration. It was fairly easy also to individualize and challenge those students who were more mature and ready for certain concepts and to provide less challenging activities for those who needed more readiness time (Durkin,1970; Papert, 1980).

The single keystroke feature of LOGO is very helpful with young children who can fairly quickly understand the use of "F", "L", and "R" as well as "D" and "U". Once children recognize these letters and the computer keyboard is color coded to provide further aid, they very easily can command the turtle to move around the screen and draw a variety of shapes and designs with little or no assistance. These drawings can then be saved and printed out with the aid of a screen dump program and a printer with graphics capabilities.

Thus, the preschoolers were introduced first to turtles in general through a live one at nursery school and some very nice puppets and stuffed ones at the Clinical Lab. Two of these, Tara (a hand puppet) and Tommy Turtle (a musical stuffed turtle) then introduced the children to the turtle language of F, L, R, U, and D. The children played games and commanded tiny ceramic and pewter turtles to go through mazes on an overhead transparency before they tried to use these commands for the turtle on the screen. They were also introduced to the computer itself, the keyboard, monitor, disk drive, floppy disks and printer. They passed around a "broken" floppy disk in order to see and feel the magnetic material that stores the information and they examined the protective covering. Soon the children were also introduced to Big Trak (Milton Bradley, 1979) a computerized tank that uses commands similar to LOGO, and to the Turtle Tot floor turtle (Harvard Associates, Inc., 1983,1984) that is programmed to move and draw pictures on large sheets of paper using either single stroke or more complex LOGO commands. They also used a floor keyboard, part of the Computergarten materials (Scholastic, Inc., 1983) on which they could either jump or place themselves, or place turtles to find various letters or numbers. A large wall keyboard chart (The Learning Circuit, 1984), color coded to help locate letters and numbers on their own keyboard, proved very helpful. Soon they were drawing and saving pictures. Some of their first attempts were merely "designs" made by pressing the command keys indiscriminately, but soon some of the children were deliberately pressing keys to make "H", a crude circle, a "pump" or triangle and one little girl had figured out how to draw a star. The children were not encouraged to try to draw anything in particular, and many continued to draw freely and "name" the picture after its creation was complete, while others drew with intention almost from the very first. While working with the

Turtle Tot during the last two weeks, the children were encouraged to estimate how far the turtle would move with steps of 1, 10, 50, 80 etc. so they began to get some relative number length associations in their minds. By the third week they moved into the more complex FDxx, BKxx, RTxx, LTxx, PU, PD, PC commands, but all in the immediate mode. These more advanced commands proved to be much more challenging for some than for others, but with assistance all could produce something on the screen by the end of the four week period. During the last three sessions the children were able to draw two separate squares side by side in a color of their choosing, and to draw a figure that resembled a barn, but even at this stage most needed considerable step by step guidance. These two figures were used to help with the concepts of 90 and 45 degree angles, which were the only ones specifically discussed.

To reinforce some of the concepts and the sight recognition of the letters, numbers, and shapes, some other commercial programs for preschoolers were used to supplement the LOGO activities. These also provided a needed change of pace. Some of the most successfully used ones were the Muppet Learning Keys (Sunburst, 1984), the Pre-reading/Counting disk (MECC, 1981) which includes several word and number games, the Stickybear series of ABC's Numbers, and Shapes (Optimum Resource, Inc., 1984) and Kindercomp (Spinnaker, 1983).

As often happens when a project like this one gets underway, the word gets out and soon one of the kindergarten teachers in the local schools wanted to bring in her class and their fourth grade computer "partners" for a morning of LOGO activities. So, the Lab staff put together a morning of five different activities designed to introduce the kindergartners and fourth graders to LOGO and to the capabilities of the computer. In order once more to provide a sequence of activities, moving from concrete to more abstract, a series of five stations or learning centers were planned, keeping in mind the idea that the fourth graders could be very helpful to the kindergartners. Three of these centers provided the sequence and the other two provided related activities so the logistics of the sequence would work. The kindergartners and their fourth grade partners were divided into three groups and each group progressed through the five stations spending approximately 15 minutes at each station. The sequenced stations were as follows: 1)introduction to F,L,R commands and a variety of body movement and floor keyboard activities, 2)work with the Turtle Tot floor turtle making it draw designs on paper on the floor, each pair of partners having an opportunity to push the keys on the computer and give the turtle the commands suggested by the rest of the group, and 3)hands on experience in the computer room, each pair to a computer, using the single keystroke commands to draw designs, shapes, and pictures on the screen. The other two stations consisted of 1)making walnut shell turtles to take home, and 2)singing turtle songs written to familiar tunes such as "Mary Had a Little Turtle" and "I've Been Working with the Turtle" and listening to tales of Yertle, the Turtle and The Tortoise and the Hare.

Evaluation of the two projects was formative as well as summative. Plans were revised from day to day and often within the day itself as a flexible and alert group of teachers noted the need for change. The major goals of discovering 1)what children of this age could actually do, 2)the range of abilities and attention spans for work within this age group as applied to computer usage, 3)whether the children would enjoy and benefit from the activities planned, 4)whether the children would make significant progress with the various concepts of right, left, 90 and 45 degree angles, etc., and 5)whether it would be reasonable to involve our early childhood preservice teachers in the future, were all met, though measured purely by observation in this pilot project.

Although there are definitely some things to approach differently the next time, both the pilot project and the one day "spinoff" project were deemed a success. Already the kindergarten teacher involved in the "spinoff" project is making plans to order LOGO; other teachers in her building will also want to try it and soon some of these teachers will be exploring the use of a programming language that can help with logical thinking and problem-solving skills. Thus, planning for "downtime" use of facilities between semesters can prove to be an effective technique for piloting an innovative program that can later be implemented more efficiently and effectively. Such programs will lead to similar cooperative projects using preservice teachers and their college of education instructors as personnel and in this way, preservice and inservice teachers will be enabled to instruct more effectively using computers.

REFERENCES

Abelson, H.,&DiSessa,A.(1980)Turtle Geometry. Cambridge,MA: The MIT Press.

__________(1979). Big Trak(Computer Device).Springfield,MA:Milton Bradley Company Electronics.

__________(1983).Computergarten(Computer Device and Program). New York,NY: Scholastic, Inc.

D'Ignazio, S. Frederick.(September 1980). The World Inside the Computer. Creative Computing Magazine.

Durkin, D.,(1970).A Language Arts Program for Pre-First Grade Children: Two Year Achievement Report. Reading Research Quarterly,5, 534-565.

__________(1983,1984).Turtle Tot(Computer Device and Program). Somerville,MA: Harvard Associates, Inc.

Henson Associates, Inc. Trademark(1984). Muppet Learning Keys(Computer Device and Program). Pleasantville, NY: Sunburst Communications.

__________(1984).Keyboard Chart Apple IIe (Computer Device). Los Angeles,CA: The Learning Circuit.

__________(1983).Kindercomp(Computer Program). Cambridge,MA: Spinnaker Software Corporation.

Moursand,D.,(December 1977). Report of the ACM Teacher Certification Subcommittee,SIGCSE Bulletin,9(4), 8-18.

Optimum Resources, Inc. Trademark(1984). Stickybear ABC's(Computer Program). Distributed by Weekly Reader Family Software.

Optimum Resources, Inc. Trademark(1984). Stickybear Numbers(Computer Program). Distributed by Weekly Reader Family Software.

Optimum Resources, Inc. Trademark(1984). Stickybear Shapes(Computer Program). Distributed by Weekly Reader Family Software.

Papert,S.,(1980). Mindstorms: Children, Computers, and Powerful Ideas. Basic Books,Inc.

__________(1981).Pre-Reading/Counting (Pre-2)Vol.7,Version 1.0 (Computer Program). Minnesota: MECC.

Rieber,L.,(1985). The Effect of LOGO on Young Children: A Research Study. Paper presented at the 26th International ADCIS Conference, March 26-28, 1985, Philadelphia, PA.

Watt,D.(1983). Learning With Logo. New York: McGraw-Hill, Inc.

THE USES OF CBE IN TEACHING
THE FUNCTIONS OF COMPLEX SYSTEMS

JOEL A. MICHAEL AND ALLEN A. ROVICK

RUSH MEDICAL COLLEGE, CHICAGO, IL

Abstract

To understand the functions of complex systems (as are found in physiology) means that one is able to predict the behavior of the systems or can solve problems about those systems. This requires not only the memorization of "facts" but the integration of those facts into conceptual schema. As teachers of medical physiology it is our responsibility to assist students in acquiring both "content" (facts) and "process" (problem solving skills). Mathematical models of physiological systems can be used in a variety of computer-based exercises to assist students in attaining the required level of understanding of the systems of the body. Such exercises include: (1) demonstrations used in lecture or tutorials, (2) computer simulated experimental preparations used in faculty directed "laboratories", and (3) computer simulated experiments in which both the physiological preparation and the instructor's didactic feedback is simulated.

The Difficulties of Teaching Physiology

Physiological systems are "complex" in several ways. (1) Every system is made up of many macroscopic component parts having specific properties. (2) Each of the components is in turn made up of tissues and cells in which many different processes occur at the microscopic and molecular levels. (3) Components at every level interact with one another in complex ways, frequently involving negative feedback (and occasionally positive feedback). (4) Like all real systems, physiological systems exhibit dynamic behavior, although there is the added complication that living systems exhibit a very wide range of time constants (microseconds to years).

As teachers of physiology it is our responsibility to help our students understand the functioning of these complex systems. The difficulty is that in such systems " ... the whole is more than the sum of its parts ... in ... that, given the properties of the parts and the laws of their interaction, it is not a trivial matter to infer the properties of the whole" (Simon, 1962). It is certainly "not a trivial matter" for physiology students to integrate the "facts" about the parts into a useful schema for understanding the function of the whole system.

Traditionally, we have utilized several teaching methods to help students master physiology. Lectures and textbooks have been our primary teaching tools to organize and convey the "information" that is required for a

BEST COPY AVAILABLE

mastery of our discipline. Small group tutorials have been used as a means of reviewing student understanding, and particularly as a medium for assisting the student in problem-solving. Laboratories have been used to achieve a wide variety of goals (Michael and Rovick, 1984). Importantly, they also afford opportunities for active problem-solving experience (analyzing the outcome of experimental procedures) and for observing the many time-dependent, simultaneous phenomena occurring in an animal.

Most of the traditional teaching methods employed in physiology are basically static, and for the student the experiences are passive ones. At best a sequence of fixed "images" can be used to convey the complex behavior of a system. However, limited time and high production expense restrict us to using at most a few such graphic illustrations.

Laboratories are clearly the best of the traditional media for helping our students to learn to deal with the complexity of physiology, but their use is limited by their expense and the time they require in the curriculum. Both of these problems have led to a well documented decrease in the number of laboratory exercises in medical physiology courses (Rothe, 1983).

As a result, our students are most successful at memorizing those definitions, concepts, facts etc. that we indicate are required to pass the course, and least successful at applying those facts and concepts to the solution of real problems or the construction of useful mental schema of the phenomena to be understood (Eichna, 1980; AAMC, 1984).

Computer-Based Mathematical Models

The existence of mathematical models describing the function of physiological systems, and the ready availability of digital computers on which these models can be programmed to run, has provided physiology teachers with an additional tool with which to assist students in mastering the complexities of physiology.

The ease with which such computer-based models can be manipulated makes it possible to use them to achieve a number of useful things. (1) Time can be compressed or expanded, thus allowing more convenient observation of the phenomena of interest regardless of their time constants. (2) The user can actively manipulate one or more of the system variables and observe the simulated system behavior. (3) As many or as few of the multiple input and output variables of the system as are of interest can be observed simultaneously. (4) The structural or functional "organization" of the system can be directly visualized, and the consequences of altering that organization (whether "experimentally" or in the presence of "pathology") can be studied. (5) Simulated experiments, conceptually identical to those traditionally carried out in the student laboratory, can be performed. (6) Systems can be observed, or experiments performed, that can not normally be made available to students, or which may not even be possible in real life.

Computer models can be used in the teaching/learning of physiology in three ways.

Simulations can be used in a demonstration mode, thus making visible the time-varying interaction of the multiple variables that describe the system. This mode is most often employed in a classroom (lecture) setting with the usual passive audience. Used in this way, computer models have been referred to as "smart video slides" (Hogan, 1982). However, the ready availability of computers also permits individual student use of such tools. In either case, the user is likely to remain a more or less passive observer of the CRT screen, certainly not a situation leading to maximal learning.

Mathematical models can also be used as simulated physiological preparations with which experiments can be carried out. Used in this mode, the instructor usually provides a laboratory manual that directs the student to follow a particular protocol and poses questions for the students to answer. While such simulations may permit the student to carry out experiments that can not, for many reasons, be done on actual biological preparations, these experiences too can remain passive ones, with the students "cook booking" the experiment and probably learning little from it. The learning that does take place is usually delayed to the time when the questions in the lab book must be answered or the lab report written. The lab itself may have provided data but generated little or no learning.

Most effective is the use of models in the context of an interactive, didactic exercise in which the student is guided by a program through a set of experiments. Used in this way, the student can be continually challenged to think about and understand the physiological phenomena being studied. He can be asked to make predictions about the outcomes of various steps before being allowed to carry out those procedures on his model. This represents the most active form of learning, one which, in our experience, makes possible the "higher order" learning about physiological systems that we want our students to achieve.

Computer Simulations Used in Teaching Physiology

We will now describe several examples of computer-based simulations of physiological systems that are used in teaching physiology.

Demonstration Models

Two examples of the use of simulations as lecture aids will suffice to illustrate the benefits of such applications.

The Hodgkin-Huxley model of membrane excitability (Hodgkin and Huxley, 1952) is of moderate mathematical complexity (a system of differential equations), and the number of parameters involved is large enough so that it is difficult to visualize the operation of the mechanisms postulated. Furthermore, experiments to demonstrate these relationships are beyond the capability of almost all student laboratories. Thus, this model is usually presented through the use of a static graphs of the pertinent variables. A computer realization of this model by Randall (1980) can be used to demonstrate all of the important properties of membrane excitability. That is, parameters can be varied and the consequence of such changes immediately visualized. Furthermore, an event that takes at most a few milliseconds in real time can be observed with graphical and tabular outputs over any time span that is required. In fact, the simulation process may be temporarily halted at any time for closer inspection of the individual variables. This is not possible in an actual experiment.

Another physiological phenomenon that has traditionally posed serious problems for students is the relationship between venous return to the heart and cardiac output. The conceptual difficulties are that each parameter determines the other and there are unequal time delays in this interaction. This behavior is difficult for students to grasp. As a consequence, they are unable to use this relationship to predict changes that will occur in the system following perturbations to either variable. A computer model of this aspect of cardiovascular function (Peterson and Campbell, 1984) demonstrates, in a more or less concrete way, the experiments from which this relationship was first established. Again, the program can be used interactively, allowing the lecturer (or students) to vary parameters and observe the consequences. The superb color animation employed, as well as the possibility for user interaction with the model, seem to increase the learning that occurs.

Simulated Experimental Preparations

A large number of models have been written to be used as "simulated preparations" on which experiments can be performed.

Modell etal. (1983) has generated a number of small models each dealing with one or a few essential respiratory relationships. While these relationships are generally described by simple algebraic equations, students require considerable practice in order to be able to solve problems and to develop a "sense" of what the expressions mean. These programs are accompanied by a laboratory manual containing suggested protocols, tables to be completed and graph paper for entering data. There are also questions to be answered. Working through the total set of programs will give the user the chance to individually study most of the major components of the respiratory system.

While the models developed by Modell are "small scale" ones dealing with more or less isolated respiratory phenomena, Coleman and Randall (1983) have adopted a much different strategy. They have described a comprehensive model of a substantial part of physiology called HUMAN. With this model not only is it possible to study the interactions occurring at the level of a single organ system, but the interactions between organ systems can also be observed. This program, too, is accompanied by a detailed laboratory manual suggesting specific protocols to be attempted.

Simulated experimental preparations of the kind described here suffer from two significant limitations. First, the user can choose to remain essentially passive, interacting with the program only in a limited way to enter parameter values or to respond to the "prompts" appearing on the screen. This is, of course, the well known "cook book" approach to student experiments that has disturbed science teachers for years. The answer is to have available an adequate number of instructors to challenge the students with questions and to point out relevant "events" as they occur. However, even the most skilled teacher can not always accomplish this with all students. And a student working independently (as is increasingly feasible with the wide spread availability of personal computers all over campus) has no such guidance available. Since one of the advantages of the computer simulated experiment over the traditional scheduled student laboratory is the possibility for ad lib use when the student is ready for the experience, this is a second, and most serious limitation. The result, then, is likely to be less learning than would be desirable.

Computer-Simulated Experiments

If the mathematical model of the physiological system is made part of a program which guides the user through a set of protocol steps and interacts with the user to explicate the physiology being taught, the result can more properly be call a "simulated laboratory exercise". That is, both the physiology and the usual input from an instructor are simulated by the program.

We have developed several such simulated experiments. The simplest of them is called LENGTH-TENSION, and it simulates a traditional student laboratory experiment on the mechanical responses of skeletal muscle (Michael and Rovick, 1984). While the phenomenon of interest does not appear to be a complex one (even if its molecular explanation is), experience has shown that students have significant difficulty understanding and applying it correctly. The model on which the simulated experiment is based consists of three simple equations and a parameter look-up table to generate the required physiological responses (Michael and Rovick, in press).

The simulated experiment contains several features that warrant some description. The program first interactively establishes that the student understands both the experimental apparatus to be employed and the procedures to be carried out. It then allows the student to begin collecting data. However, after three data points are collected it asks the user to predict the outcome of the fourth stimulus before allowing the user to see the results. Any error in prediction is discussed after the result is available. When the student has generated two consecutive correct predictions this routine is halted. Following collection of enough data to describe the length-tension relationship, an interactive dialog pursues significance of the relationship found and the molecular mechanisms that give rise to it.

This approach to a simulated experiment requires that the user carry out the exercise with an "active head" as well as with "busy fingers". It also ensures that all of the important objectives associated with the exercise are mastered on-line, and does not rely on the often unfulfilled promise that, "I'll look at the results back in the dorm and make sense out of them then."

A similarly formatted, but much larger scale, teaching program is HEARTSIM (Rovick and Brenner, 1983; Michael and Rovick, 1984). This simulation deals with those components of the cardiovascular system responsible for maintaining a constant blood pressure, and the model on which it is based is the well-known "MacMan" (Dickinson etal., 1973). HEARTSIM makes very extensive use of a "predictions table" in which the user must indicate the directional changes that will occur to eight parameters as a result of the particular system perturbation to be tried. Only after these predictions have been entered can the user actually run the simulation and obtain the results. The user is then returned to the "predictions table" and the simulation results are entered alongside the predictions. Any discrepancies between results and predictions are high-lighted and brief interactive dialogs are used to correct the errors in fact or concept that led to the incorrect predictions.

The system being modelled here is of more than moderate complexity, with there existing strong interactions between the parameters being monitored. Our students come to this exercise only after completing the relevant lectures and reading. Yet in spite of their initially high level of confidence, most students find it very difficult to predict the consequences of the first few procedures tried. However, it is clear that practice with a number of such problem-solving exercises improves the accuracy of their predictions and makes them much more comfortable with the complexity of these interactions.

Discussion and Conclusions

Currently microcomputers possess the computational power, speed, and ability to generate high quality graphics needed to run simulations of complex physiological systems. Such computers are also now relatively inexpensive and are rapidly becoming ubiquitous on college and medical school campuses.

Furthermore, modelling of physiological systems is proceeding at a rapid pace, assisted by the existence of national centers for such simulation activity at the University of Washington (J. Bassingthwaithe) and at Duke University (M. Kootsey).

If we add to this list a growing understanding of how problem-solving as an activity can be taught more effectively (Tuma and Reif, 1980), and an appreciation for the teaching power of computer-based interactive dialog and system manipulation (Steinberg, 1984), we have a recipe for an approach to the teaching of physiology (or any of the biomedical sciences: Michael, in press) that will enable us to stress the conceptual understanding as opposed to the rote accumulation of "facts".

References

1. Association of American Medical Coleges. Physicians for the Twenty-First Century, The GPEP Report. Washington, DC: AAMC, 1984.
2. Coleman, T.G. and Randall, J.E. HUMAN, a comprehensive physiological model. The Physiologist 26: 15-21, 1983.
3. Dickinson, C., Goldsmith, C.H. and Sackett, D.L. MacMan: a digital computer model for teaching some basic principles of hemodynamics. J. Clin. Comput. 2: 42-50, 1973.
4. Eichna, L.W. Medical-school education, 1975-1979. New Engl. J. Med. 303: 727-734, 1980.
5. Hodgkin, A.L. and Huxley, A.F. A quantitative description of membrane current and its application to conduction and excitation in nerve. J. Physiol. 117: 500-544, 1952.
6. Hogan, P.M. Computer generated figures in physiology teaching: the "smart video slide". 33rd Ann. Fall Meeting, American Physiological Society, San Diego, CA, October, 1982.
7. Michael, J. A. (editor). "Computers in the Teaching of the Biomedical Sciences: Papers from the 69th Annual FASEB Meeting, April, 1985." The Physiologist, in press.
8. Michael, J.A. and Rovick, A.A. Computer-simulated experiments in the teaching of physiology. Proc. 6th Ann. Natl. Educ. Comp. Conf., Dayton, Ohio, 1984. pp. 20-24.
9. Michael, J.A. and Rovick, A.A. Simulations used in teaching: ways to generate them. Undergrad. Microcomp., in press.
10. Modell, H.I., Olszowka, A.J., Plewes, J.L. and Farhi, L.E. Role of computer graphics in simulations for teaching physiology. The Physiologist 26: 93-95, 1983.
11. Peterson, N.S. and Campbell, K.B. Simulated laboratory for teaching cardiac mechanics. The Physiologist 27: 165-169, 1984.
12. Randall, J.E. Microcomputers and Physiological Simulation. Reading, MA: Addison-Wesley Publishing Company, Inc., 1980.
13. Rothe, C.R. Trends in physiology teaching laboratories for medical students-1982. The Physiologist 26: 148-149, 1983.
14. Rovick, A.A. and Brenner, L. Heartsim: A cardiovascular simulation with didactic feedback. The Physiologist 26: 236-239, 1983.
15. Simon, H. A. The architecture of complexity. Proc. Am. Philos. Soc. 106: 367, 1962.
16. Steinberg, E.R. Teaching Computers to Teach. Hillsdale, NJ: Lawrence Erlbaum Associate 1984.
17. Tuma, D.T. and Reif, F. (editors). Problem Solving and Education: Issues in Teaching and Research. Hillsdale, New Jersey: Lawrence Erlbaum Associates, 1980.

Substance Abuse CAI:

A Computer-Based Approach to Primary Prevention

Barbara S. Thomas, Ph.D.
Greg Saunders and Ron Roskam

The University of Iowa College of Nursing

Abstract

Statistics concerning substance abuse among the nation's adolescents have increased steadily despite many attempts at primary prevention. It has been reported that one particular drug may decrease or increase in popularity, but the overall use patterns remain high. Recent trends indicate that alcohol use is increasing while marijuana use is decreasing, and that female substance abusers are "catching up" with their male counterparts. There is firm evidence that age of onset of use is moving steadily downward. A consensus that prevention is the only intervention which can stem the tide is growing. Treatment, even when it is successful, does nothing about the damage that is done to youthful drug abusers, or the stress that their families and friends experience. To be effective, primary prevention must be aimed at elementary school students.

Primary prevention strategies have included three general types: (1) scare techniques, (2) information-based programs, and (3) values clarification. Each of these approaches has given too little attention to the tenets of social psychology and the fact that drug misuse and abuse are learned behaviors. The programs designed to deter alcohol misuse or abuse will be used as an illustration of the reasons for failure of many drug abuse prevention programs.

Scare techniques generally focus on the evils of drinking and the development of alcoholism, stressing that alcoholism is a terminal disease. While the factual basis of many components of these programs, particularly those concerning alcoholism, may be correct, the goal of total abstinence for everyone is unrealistic. Moreover, the stance of moralistic preaching about alcohol use is at odds with children's observations and experience.

Information-based programs provide children with facts about drugs without dealing with the forces which promote drug use: (1) peer pressure, (2) the effects of adult role models, (3) the influence of the media, and (4) children's unmet needs. Moreover, the inclusion of statistics about widespread use of drugs among adolescents in these programs may actually promote experimentation and use.

The values clarification programs come the closest to dealing with the real world problems that youngsters face in deciding whether or not to use drugs. Generally, a good deal of attention is placed on decision-making, and many include the use of peer counseling. Unfortunately, there is a serious shortage of teachers and counselors who can run these programs effectively. In the hands of ill-prepared leaders, values clarification can give children the notion that any decision they make is all right. Ignoring family and community norms as a result of less than competent leadership has been the main problem in these programs.

Philosophy of Substance Abuse CAI

The computer-assisted instruction (CAI) programs which make up the series, Substance Abuse CAI, are based on learning theory, principles of social psychology and the theory of cognitive dissonance. Children are given correct information about drugs, guidance in understanding why people use drugs, an overview of the metabolism of drugs in the body and the short-term and long-term effects of using different drugs. In addition, there is a strong focus on instruction geared to promote critical thinking and a rational approach to decision-making.

Opportunities to confront situations in which drug use is promoted are provided so that children can rehearse responses to these situations in advance of encountering them. In using the CAI programs, users must enter responses indicating a responsible approach to such situations in order to obtain positive feedback. If they hold irresponsible attitudes while repeatedly responding with responsible answers, their attitudes will shift toward responsible decision-making according to the theory of cognitive dissonance.

Description of the Programs

Four of the five programs which make up this series are being finished and tested with teachers, school nurses and students aged 9-20: (1) An Introduction to Psychoactive Drugs, (2) Drinking and Not Drinking: The Choice is Yours, (3) Keep Off the Grass, and (4) Six Classes of Psychoactive Drugs. The fifth program is in the planning stage; it is hoped that "Techniques of Persuasion" will be a Level 3 videodisc.

The techniques used in these programs include drill and practice, information entry and retrieval, tutorials, self-tests, graphics, gaming, problem solving and self-tests. Three include gaming and one includes information entry and retrieval. Figure 1 below, illustrates the modes used in the four CAI programs.

The menu for **An Introduction to Psychoactive Drugs** consists of the following: (A) An introduction which defines DRUG, DRUG MISUSE and DRUG ABUSE, (B) Drugs, Circulation and the Nervous System, (C) Drugs, Tolerance and Dependence, (D) Drugs and the Law, and (E) A Self-test. Graphics and animation are used very heavily in the first tutorial to depict metabolism of drugs and their effects in appropriate ways for elementary school children. For example, animation was used to show the vesicles' movement toward the presynaptic membranes.

Drinking and Not Drinking: The Choice is Yours also consists of a five part menu: (A) Drinking or Not Drinking -- A Self-Test, (B) Your Interests Please?, (C) Alcohol's Effects: A Tutorial, (D) Taking Chances: A Game of Chance, and (E) Key Terms and Definitions. Graphics are used liberally in this program to illustrate the path that alcohol takes in the body and the time that it takes for metabolism. "Your Interests Please?" gives users a chance to enter questions they have about drinking, anonymously. These questions can be printed out so that the teacher, school nurse or counselor can select the most appropriate means of answering children's questions -- a poster or a small group discussion, for example.

Keep Off the Grass uses a self-test, a tutorial, a glossary, and a game to provide learners with information about marijuana and situations which require decision-making skills. Graphics are used to explain the effects of marijuana on the various systems of the body. Information about hashish and hash oil is also included in this program.

Six Classes of Psychoactive Drugs deals with amphetamines, barbiturates, coca and cocaine, hallucinogens, opiates, and volatile inhalants. The menu for this program begins with an introduction which provides users with two systems of classifying psychoactive drugs, one based on effects produced (sedative-hypnotics, narcotics, central nervous system stimulants, and psychedelics) and the other based on the potential for abuse -- the six classes listed above, as well as alcohol and marijuana.

Evaluation of the Programs

Formative evaluation was a continuous process during the writing and programming of these programs. Content experts and instructional designers reviewed the program before programming began, and a number of changes were made. In our project, this has become the standard approach, and it should be noted that the programmers are excellent at screen design as well as being wizards at programming.

When each program had been refined, it was placed in various sites for field trials. For these programs, the sites included public elementary schools, nursing programs, participants at a school nurse conference, and workshops which we conducted for nurses in various settings, e.g., hospitals, Visiting Nurse Associations.

Discussion

To date, only one program has been located that deals even faintly with the problem of primary prevention of mood altering substances. Sunburst has advertised a program on smoking, along with another on using seat belts -- the general focus is on development of healthful lifestyles. To my knowledge, there is no indication regarding the effectiveness of these programs.

An effort is being made to arrange controlled field trials to assess long-term effects as well as short-term outcomes. Pre-testing and post-testing will take place with experimental and control groups assigned at random among comparable elementary schools -- comparable on Iowa Tests of Basic Skills (ITBS) scores and on socio-economic status. The testing will provide data about students' knowledge base and also about selected learner variables such as self-esteem and need for control. It is also hoped that a longitudinal study can be completed to learn about knowledge and attitudes during the junior high school and high school years, onset of use, and frequency of use. A difficult part of this study will be the definition of the drug education programs which already exist in the test schools. No doubt observation will be needed to supplement information gained from the curriculum outlines.

	Program			
	I	II	III	IV
Drill and Practice	x	x	x	x
Tutorial	x	x	x	x
Information Storage and Retrieval	x	x	x	x
Graphics	x	x	x	x
Self-Tests	x	x	x	x
Simulation	x	x	x	x
Gaming		x	x	x
Animation	x	x	x	x

Figure 1

Attitudes Toward a Computer Simulation to Assess Quality of Ready-to-Wear

Ruth Marshall

Iowa State University

The purpose of this research was to study student attitudes toward a computer simulation developed to assess construction quality in ready-to-wear garments. These three research questions were central to the project:

--How effective is the computer simulation in teaching evaluation of ready-to-wear?

--Which learners had the more positive attitudes toward the simulation?

--What variables are predictors of students' reactions to the simulation?

The study made use of a computer simulation, BUYER, developed for use in a basic clothing construction course at Iowa State University (Marshall, Hooper & Slaybaugh, in press). BUYER simulates for the students a buying trip in which they must decide the number of garments to purchase for a department in an imaginary retail store on the basis of the construction quality of the items.

The instrument used in the study consisted of an attitude scale and a background information sheet. The attitude scale contained 30 items which addressed attitudes toward computers in general, the evaluation of ready-to-wear, the class, and this computer simulation activity; a 9-point scale was used in the instrument. The instrument also gleaned such information about the students as classification, sex, major, amount of prior experience in clothing construction, amount of prior use of computers, and the percentage of their own clothes which are personally sewn or factory produced.

The 131 subjects of the study represented students enrolled in a basic clothing construction course at Iowa State University spring and fall semesters, 1984. Data from the questionnaires were analyzed using both descriptive and inferential statistics. Results suggest that the students were able to determine construction quality through a computer simulation. However, no clear profile has emerged of the learners who would have the more positive attitude toward this computer exercise.

Reference:

Marshall, R., Hooper, E., & Slaybaugh, J. (in press). BUYER: A computer simulation to assess construction quality in ready-to-wear garments. Journal of Computer-Based Instruction.

A Proposed Model for Field Testing

Frances Whitaker Mayhew and David M. Johns

University of Delaware

ABSTRACT

Following completion of programming and internal review, an educational program is ready for evaluation in a classroom setting. Design weaknesses and programming problems which are not discovered through preliminary evaluation can be identified through field testing. This paper presents and discusses a structural model for the evaluation of computer-based instructional materials through field testing.

INTRODUCTION

One of the final stages in the development of educational software involves testing with a group of students. Students are experts in discovering programming, instructional design and content errors that the most careful internal reviewers fail to notice. Student evaluation can be conducted in four stages as follows:

1. Students can volunteer to work through the program,
2. The author can use the program in an on-site classroom situation,
3. The program can be utilized in an off-site classroom situation by a cooperating instructor,
4. The program can be subjected to a "stand alone" trial.

Voluntary trial runs can be very helpful in the early stages of development to catch major problems. These usually operate informally with students relaying any useful comments to the author or programmer. On-site classroom evaluation permits the instructor/author to observe the integration of the instructional software into its intended setting. Students can be depended upon to report problems enthusiastically, especially if they fear that their grades may be effected. The addition of some structural components to the evaluation in the later stages of development can provide more specific information. This information can then be used to refine the program by pinpointing trouble spots. Off-site classroom evaluation should not be attempted prematurely. The author and his institution run the risk of unauthorized publication of unflattering reviews and a general negative impression on students and professional colleagues alike. In the final preliminary trial, the program is utilized in a classroom setting without the supervision of the developer. In essence, the program is required to stand on its own merit.

OBJECTIVES

The objectives of the following proposed model for field testing are as follows:

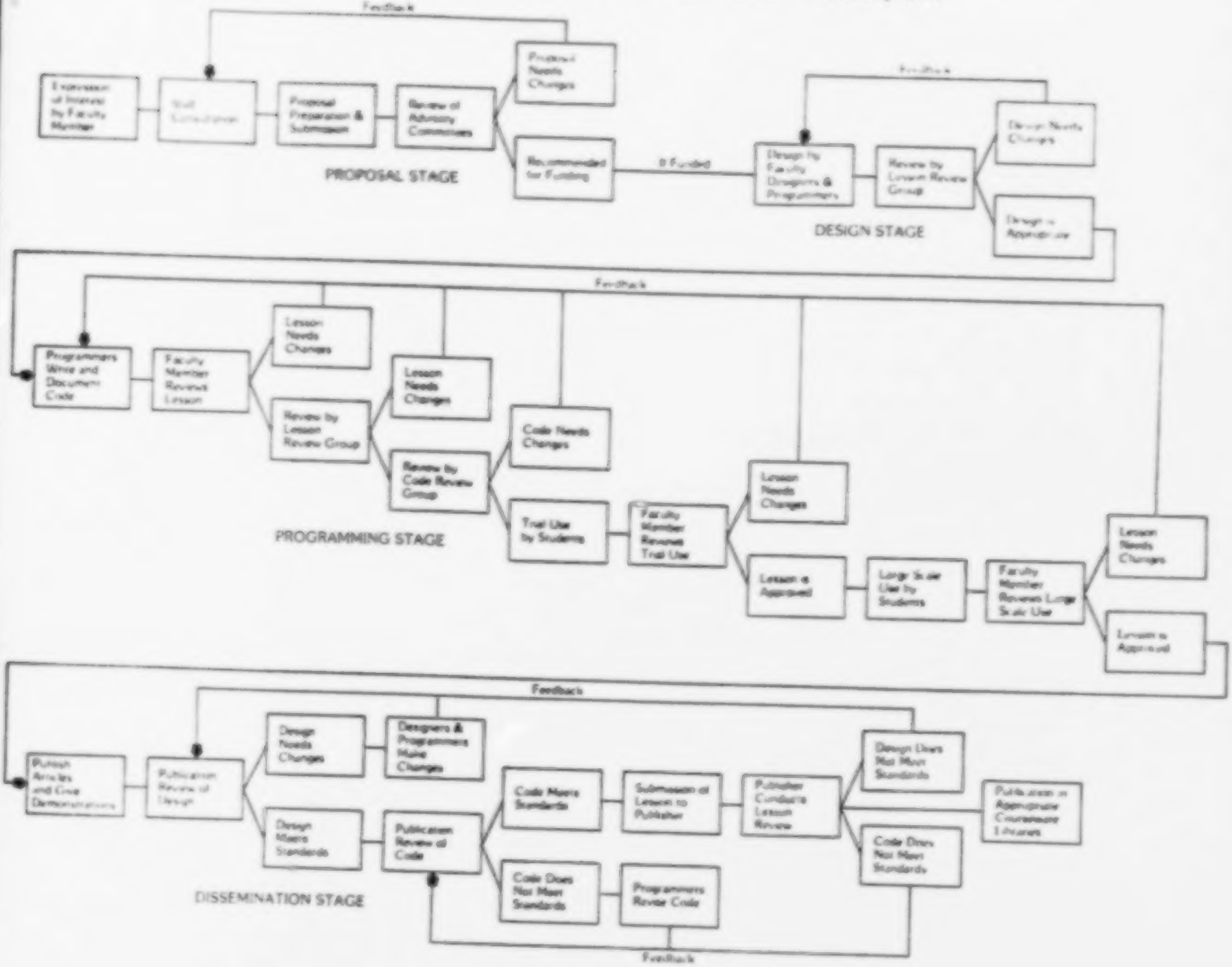
1. To refine and generalize the instructional package,
2. To take full advantage of the innate objectivity of the classroom situations, and
3. To insure that the instructional package is reliable enough to access student mastery.

PROPOSED MODEL

The structural format for this model was suggested by the diagram of The Delaware Model for Courseware Development (Seiler, 1981). The Delaware Model

Figure 1

The Delaware Model for Courseware Development


Feedback
Expression of Interest by Faculty Member
Staff Consultation
Proposal Preparation & Submission
Review of Advisory Committees
Proposal Needs Changes
Recommended for Funding
PROPOSAL STAGE
If Funded
Feedback
Design by Faculty Designers & Programmers
Review by Lesson Review Group
Design Needs Changes
Design is Appropriate
DESIGN STAGE
Feedback
Programmers Write and Document Code
Faculty Member Reviews Lesson
Lesson Needs Changes
Review by Lesson Review Group
Lesson Needs Changes
Review by Code Review Group
Code Needs Changes
Trial Use by Students
Faculty Member Reviews Trial Use
Lesson Needs Changes
Lesson is Approved
Large Scale Use by Students
Faculty Member Reviews Large Scale Use
Lesson Needs Changes
Lesson is Approved
PROGRAMMING STAGE
Feedback
Publish Articles and Give Demonstrations
Publication Review of Design
Design Needs Changes
Designers & Programmers Make Changes
Design Meets Standards
Publication Review of Code
Code Meets Standards
Code Does Not Meet Standards
Programmers Revise Code
Submission of Lesson to Publisher
Publisher Conducts Lesson Review
Design Does Not Meet Standards
Code Does Not Meet Standards
Publication in Appropriate Courseware Libraries
DISSEMINATION STAGE
Feedback

BEST COPY AVAILABLE

identifies the process through which the Office of Computer-Based Instruction at the University of Delaware supports the development of educational software. A team approach is used. The team consists of a faculty member who provides the content expertise and develops the initial program script, a programmer who commits the script design to code; and a group of instructional design experts who periodically review the program. The proposed Mayhew/Johns Model fits into the Delaware Model between the "Trial Use by Students" component and the "Faculty Member Reviews Large Scale Use" component. Essentially it is an expansion of the student-related evaluation activities presented in the Delaware Model. The Delaware Model is presented in Figure 1. Since field testing is usually an instructor-initiated activity, the Mayhew/Johns Model attempts to mimic the process of a "lesson development" strategy. This strategy employs successive trials with intervening revisions which are generated though the evaluation of each trial. The Mayhew/Johns Model is presented in Figure 2.

The Mayhew/Johns Model identifies three components which follow parallel paths of development. The upper line of the Model shows the development of the tools used to collect evaluative information related to the instructional software. The middle line of the Model indicates the steps in the development of the testing situations. The lower line of the Model identifies the development of the instructor's manual which documents the program.

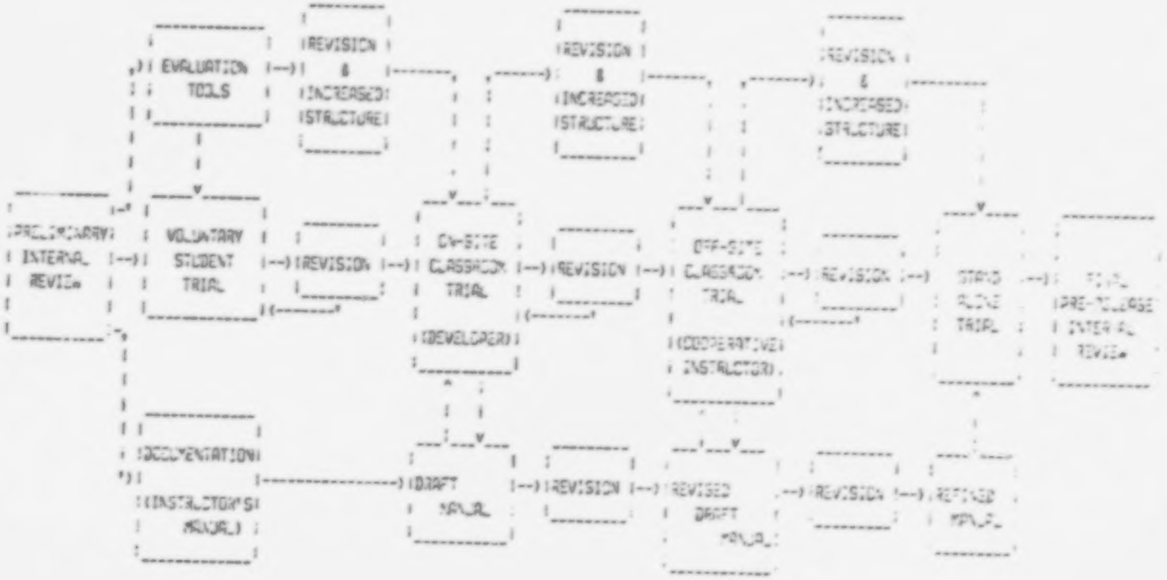
EVALUATIVE TOOLS

Three types of evaluative tools are suggested to gather information during the testing situations--a structured student log, a pretest/posttest and subjective reviews. These tools should be designed to identify three types of structural weaknesses--programming weaknesses, content weaknesses, and weaknesses in the instructional design. Programming weaknesses are problems in the coding which cause student frustration. Content weaknesses include narrow or incorrect interpretations of theory. Problems with the instructional design will introduce confusion and short-circuit the effectiveness of the program.

Multiple operations on a programmable function key such as CONTOL-b or F1, operations which are not consistent with explanations in the instructions; or an infinite loop in the code are examples of programming weaknesses. Such "bugs" in the program are often easily remedied but in many cases difficult to isolate. An evaluative tool in the form of a structured log which directs the

FIGURE 2

MAYHEW/JOHNS MODEL


PRELIMINARY INTERNAL REVIEW
EVALUATION TOOLS
VOLUNTARY STUDENT TRIAL
REVISION & INCREASED STRUCTURE
REVISION
ON-SITE CLASSROOM TRIAL (DEVELOPER)
REVISION & INCREASED STRUCTURE
REVISION
OFF-SITE CLASSROOM TRIAL (COOPERATIVE INSTRUCTOR)
REVISION & INCREASED STRUCTURE
REVISION
STAND ALONE TRIAL
FINAL PRE-RELEASE INTERNAL REVIEW
DOCUMENTATION (INSTRUCTOR'S MANUAL)
DRAFT MANUAL
REVISION
REVISED DRAFT MANUAL
REVISION
REFINED MANUAL

BEST COPY AVAILABLE

student to indicate specifically his location in the program when encountering the problem can assist the programmer in correcting it.

Meaningless feedback to test item responses, insufficient tolerance in correct answers; or conceptually inaccurate textural material are examples of content weaknesses. The structured student log can be useful in locating this type of problem. In addition, a pretest-posttest instrument may be useful in determining content-related problems which will weaken the teaching effectiveness of the program.

Problems in the instructional design of the program include poorly designed screen displays, confusing instructions or test items which do not give the student the correct answer after repeated incorrect responses. This type of problem is more difficult for students or a cooperating instructor to pinpoint than programming or content weaknesses, however the structured log response sheet can direct the instructional designer to specific test items or screen displays which may need improvement.

The Mayhew/Johns Model indicates that each successive step in the refinement of the instructional package will probably suggest revisions which should be made in the evaluative tools. As a result, the evaluative tools will become increasingly more structured as the evaluation process continues.

The inclusion of subjective reviews from students and the cooperating instructor provides a vehicle for this group to make suggestions or comments which might otherwise be missed.

In addition, many computer systems have the capability to gather useful student data such as the number of times a student tries a test item before producing a correct answer, or retaining unanticipated wrong answers. Sometimes "unanticipated wrongs" are actually correct and should be added to the list of answers among which the computer tries to find a match for the student's response.

TESTING SITUATIONS

During preliminary software development, it is often useful to have students try the program on a voluntary basis. Voluntary student reaction is often revealing and students reward the non-threatening (non-graded) situation with valuable objective evaluations. Some software developers rely on student reviewers to recommend familiar examples for theoretical concepts. Dr. James Morrison reports that a student reviewer suggested he use "beer and pretzels" as the alternate commodities in a program on Satisfaction Units in his "Consumer in the Marketplace" package rather than the more traditional example he had originally planned (Morrison, 1980). These voluntary student trials are usually conducted informally, but structured log response sheets can be used in this situation to maximize response utilization. In addition, the observation of students as they work through an untested program can provide much useful information.

Once the total script design has been committed to code and received a careful internal reviewed as indicated by the Delaware Model (see box labeled "Review by Code Review Group"), the developer/faculty person usually begins the On-site Classroom Review phase. This step involves integration of the instructional package into a course curriculum. The instructional package is permitted to function in the curriculum according to its design, either as a integral or supplemental component. The students in the class are usually not permitted to choose whether or not they participate in the evaluation. Student mastery of content may be dependent upon their performance on the test items of the instructional package. The developer/faculty may opt to phase in the significance of the instructional package in relation to student grades by initially offering "pass/fail" grades for participation, with a passing grade awarded for completion of the package regardless of student performance. Once the instructional package has had several on-site classroom trials and major programming and content weaknesses removed, student mastery can be determined by performance on tasks or tests within the package. If student grades are not dependent upon participation in the evaluation, students will not be highly motivated to utilize the package. The structured log response sheets as well as administration of a pretest/posttest will be useful in evaluating the ease of use and effectiveness of the package. Each subsequent on-site classroom trial will reveal additional evaluative refinements. These can be incorporated into the program through periodic revisions.

Once the on-site classroom trials, under the supervision of the developer, are producing desirably reliable and positive evaluations; it is appropriate to locate a co-operating instructor to conduct an off-site classroom trial. The off-site classroom trial can be carried out in a similar fashion to the on-site trial utilizing structured log response sheet and pretest/posttest evaluation strategies to judge ease of use and instructional effectiveness. Since the student group must be similar in makeup to that participating in on-site trials, the major difference in the off-site trial is the familiarity of the instructor with the content and instructional approaches of the package. Hopefully, input from the co-operating instructor will support and strengthen the generality and transportability of the instructional package. Close informal contact should be sustained between the developer and the co-operating instructor in order to monitor the progress of the trial. Also the developer can answer questions as they arise and provide support to the co-operating instructor. A detailed subjective evaluation by the co-operating instructor as well as the student evaluations will provide substance for further refinement of the instructional package.

The final evaluative classroom trial is conducted in a "stand alone" situation. As the name implies, the stand alone trial proceeds with no interaction between the developer and the co-operating instructor. This step identifies the capability of the instructional package to function without external support. In this situation, the individual student log responses can be omitted. Pretest/posttest evaluation, subjective comments from the co-operating instructor and student subjective evaluation will confirm the readiness of the instructional package for publication release or indicate the need for further revision.

Following this series of classroom trials, each contributing to the general utility and independent application of the instructional package, a final internal evaluation is conducted by the Review Group in anticipation of publication and dissemination.

DOCUMENTATION/INSTRUCTOR'S MANUAL

The third line of development on the Mayhew/Johns Model relates to the supportive documentation which will accompany the instructional software after publication. This Instructor's Manual should include a discussion of the instructional philosophy supporting the instructional package and samples of screen displays. In addition, all the test items and tutorial questions in the program with the feedback accompanying each possible response should be provided in the Instructor's Manual. This documentation should develop along with the software package, benefitting from the evaluation of each successive classroom trial. A Draft version of the Instructor's Manual is developed as the on-site classroom trial is being conducted. This Draft version is refined to generate a Revised Draft Instructor's Manual which will be utilized by the co-operating instructor during the off-site classroom trial. Further revision based on the input of the co-operating instructor will ultimately produce the Refined Instructor's Manual. This document is used with the "stand alone" trial and will accompany the published package.

The Mayhew/Johns Model for Field Testing is offered to the community of instructional software developers for scholarly critique. The authors plan to report on specific components of this model in subsequent publications.

REFERENCES

Morrison, James L., "Project Disco: A PLATO Learning System in Consumer Economics." *1980 Conference Proceedings*. Bellinghan, Washington:Association for the Development of Computer-Based Instructional Systems, p.220-223.

Seiler, Bonnie Anderson. "Applying a Systematic Approach to CBE Development in an Academic Environment." *1981 Conference Proceedings*. Bellingham, Washington:Association for the Development of Computer-Based Instructional Systems, p.305-310.

FUTURE GROWTH AND DEVELOPMENT OF MICROCOMPUTERS IN HOME ECONOMICS

Louise Wesswick

Home Economics Department, University of Wyoming

ABSTRACT

Trends in the growth and development of microcomputers in home economics are discussed as they relate to the transition from an industrial to an information society. An increase in computers, changes in the equipment, the effects of computers on home and family, and future directions are described. Home economists are challenged to develop software in subject matter areas and to use software for record keeping and administrative purposes.

FUTURE GROWTH AND DEVELOPMENT OF MICROCOMPUTERS IN HOME ECONOMICS

As home economics participates in the transition from an industrial to an information society, there will be a great increase in the number of computer users, changes in cost, structure, and performance of the equipment itself, and a significant impact on the home and family in terms of life styles, education, and occupations available.

Increase in Computers

We are today surrounded by millions of computers, and we are told that their increase in the last 25 years is small compared to what we are yet to see.[1] By the year 2000, ninety-five per cent of all schools and businesses will have computerized ninety-five per cent of their procedures.[2] Education is relying increasingly on computers as an educational tool. The National Center for Education Statistics has reported that by the spring of 1982, the number of microcomputers used by students for instruction had grown to almost 100,000 tripling the number available just 18 months earlier. Computers are being introduced most rapidly in senior high schools, where 74% reported they were providing computer based instruction in 1982.[3] Schools around the nation are beginning to realize that in the information society, two required languages will be English and computer.[4] As a part of today's school reform movement, the state of Oregon is adding computer literacy to the list of skills students will be required to master in high school, and Northview High School in Grand Rapids, Michigan, has also added a computer literacy course.[5] A growing number of universities require graduates to demonstrate the ability to write a simple computer program. Within the next five years, students entering college will have computer skills. In the future, students will be able to access global data banks in their field of study.

Changes in the Equipment

In the next five years, computers will provide information processing for individuals or groups who previously could not afford them or were unable to cope with their complexity. Such computers will appear in cars, home furnaces, and stores surprising people with their simplicity and usefulness. It is estimated that 5.2 million desk-top computers will be installed in offices in the near future.[6] The cost of information-processing machines will be coming down very rapidly. Various estimates place the low end to the price range for microcomputers from $150 to $500.

Computers will be smaller in size, more powerful because of added memory, and contain built-in peripherals like modems and printers. They will be easier to use (user friendly) and more portable. The combining of television, radio, telephone, and computer network will result in a home set resembling a standard television with a keyboard attached--a microprocessor that turns the set into a computer terminal as well. Later models may include a small television camera and a microphone so that video telephone calls become a part of the system. If a printer is included, the user can obtain "hard copy". Keyboards will soon appear with plain English commands like "copy" and "load" program. A most recent research report indicates that work is underway to replace the silicon in microcomputer chips with a new substance (Galium Arsenine) to make the machines virtually instantaneous in their response, functioning six times faster than at present. This process is predicted to become a two-billion dollar business by 1990.[7] Using protein in computer chips to allow billions of times more information to be contained in the same space is also being studied.

Soft and hard disks will be replaced by some form of laser video-disc technology. Before 1990, laser videodiscs will be available that will be capable of storing the entire contents of the _Encyclopedia Britannica_ on a single videodisc with room left over to write and edit _War and Peace_. Before the century is over, home computer owners will be able to display not only data and graphics, but full moving video images on anything from their walls and ceilings to tiny screens worn on the wrist.[8]

---

1. Lias, Edward J., _Future Mind_, Little, Brown, and Company, Boston, 1982, p. 247.
2. Ibid., p. 259.
3. _Rock Springs, Wyoming, Daily Rocket-Miner_, June 14, 1984.
4. Naisbitt, John, _Megatrends_, Warner Books, Inc., New York, 1982, p. 34.
5. _Rock Springs, Wyoming, Daily Rocket-Miner_, July 5, 1984.
6. Lias, _op. cit._, p. 263.
7. _CNN News_, June 14, 1984.
8. _Spinrad_, Norman, "Home Computer Technology in the 21st Century" Popular Computing, Vol. 3, No. 11, Sept., 1984, p. 77.

In the year 2000, the home computer user, via the Net Console, will have direct video and voice access to anyone anywhere in the world and will have immediately available the sum total of all human art and knowledge. The Net Console will have controllers such as conventional typewriter-style keyboards, musical keyboards, microphones, video cameras, videodisc recorders and biofeedback devices. Robots will be interfaced with computers to include a set of programs and peripheral devices which enable the program to manipulate material objects in the real world.[9] In the next ten years, robots are expected to get into the automobile and drive it. The destination is fed into the computer, and instructions are given to explain the best route to take, where the parking spaces are, how fast to go, and whether or not to change lanes. The robot then follows these directions to reach the destination.

Home and Family

Futurist writers who have examined the information revolution precipitated by the increase in microcomputers are in agreement in predicting that the most significant impact of this growth will focus on the area of the home and family. In the Third Wave, Alvin Toffler[10] coined a new term, "the electronic cottage", to identify major changes likely to occur. The "cottage industry" idea of old contained most activities within the cottage. The electronic home, likewise, will contain computers, robots, and videodiscs to be used by household members within the cottage. When the cost of telecommunications equipment drops below the cost of commuting, traditional boundaries between work and home will disappear as the future home becomes an institution in which millions of people will work at home rather than in the office or factory. In the 1990's one out of every five workers will be working at home as computers and videotex equipment replace the typewriter of today. Parents and children as a unit, making decisions together, will work at terminals earning income, as well as pursuing hobbies and absorbing information. An advantage of the electronic cottage may be that it integrates the young people into the world of work and might be considered a solution to high youth unemployment. Attractions of such a home are opportunities offered parents to spend more time with their families and to choose theirown work hours. Also, when people remain at home, the community tends to become more stable than otherwise might be the case. Networking of microcomputers from home to home may establish an electronic expanded family.

In the kitchen computer center equipped with a printer, the shopping list will be printed from the kitchen computer's knowledge of the needed items for regular meals and for future special activities. This inventory database for the kitchen computer is updated regularly, and the kitchen computer may call in the grocery store order, just as grandmother did in the old days.

Equipment in the freezer storage center will be designed to select a chosen meal and transport it to the combination convection-microwave oven, where it will be cooked to perfection. Appliances will be switched on and off as needed. Household members can tell the toaster, range, and television set what they want, and those appliances will perform accordingly. For example, one might say, "I want to watch Channel 12 at 8:00 p.m., but store the show for next week," and it will be done. Computers at the soft drink machine might talk back saying, "You didn't deposit enough cash." One robot on the market can vacuum the carpet, but it is still less expensive to buy a vacuum cleaner and run it yourself. Some companies make voice output devices that permit the computer to issue preprogrammed messages. A speech-equipped home might issue a warning when appliances malfunction or when it senses danger from intrusion or fire. A typical use of speech capability is in security systems where the computer can telephone for help in an emergency. Microcomputers may eventually replace the telephone.

A basic lighting and appliance-control system with simple thermostat setback capabilities and passive security features costs about $500. Assuming that you already have a home computer, adding computer control or other options would increase the cost to $1,000. Installing state-of-the-art automation with everything except robots would cost approximately $5,000.

A home computer can be effective in providing the home owner with safety and security. For example, should a burglar enter the home while occupants are away, the home computer will detect the intrusion by comparing the suspicious or unusual movements to the family's normal living patterns. To frighten the burglar,

---

9. Ibid., p. 78.
10. Toffler, Alvin, The Third Wave, Bantam Books, March, 1980, pp. 194-207.

the system will use speech synthesis via intercom and turn on lights while dialing police. In the event of a fire, a home computer with sensors for smoke and fire will easily protect the owner and contents of the dwelling.

There are occasions on which the home computer can be a special convenience to the home owner. For example, when plans change and the occupant wishes to return home early, he can call ahead, and the computer will adjust the lights and temperature and will open the garage door. On other occasions photo cells can tell the computer to turn on driveway lights and the sprinkler system at dusk. Temperature sensors will indicate when to turn on fans and open windows or control the environment in a room.

If a home occupant is handy with tools, a home computer will advise him on parts to purchase and lead him through the steps of repair.[11]

Combining the television set with the home computer will create a new mode of shopping behavior known as telebuying--the purchase of goods and services through using the telephone, interactive cable television, or other forms of electronic communication. Across the country, 24 hours a day, television stations will be offering for sale a multitude of products. The consumer will purchase the product by mail or phone in the conventional way or directly through an adaptor connected to the television set. Telebuying is predicted to take a firm hold by the 1990's.

Instead of strolling through malls, families may view and discuss products seen on their television sets, leading to increased group decision-making within the family and more comparison shopping. Some television stations plan to run product test results along with advertisements, making greater amounts of information available to consumers. In addition to the convenience of telebuying, market professionals believe one of its advantages is the opportunity for global interaction. No longer limited to their local shopping center, consumers will be able to buy directly from both national and international firms from all over the world.[12] While some writers are asserting that the annual Christmas rush could become a tradition of the past, there is some feeling that "computer buying will never replace the serendipity and high touch of shopping for what we want to be surprised about."[13]

An experimental project revealed some possible future uses of the microcomputer for elderly persons living alone who face special problems of isolation. Computer networks linked the elderly to each other and to three neighborhood community centers. The linkage permitted these older people to exchange ideas about mutual problems and allowed them to contact friends, relatives, government, and social welfare officials. In a follow-up study, the elderly reported that the programming had reduced isolation and served as a source of needed human contact. Approximately 80% of the participants stated that the interactive cable television system contributed to a major improvement in their lives.[14]

In the 1990's the computer will serve as an adviser, housekeeper, secretary, and friend to the elderly. Biofeedback monitors attached to the clothing of the elderly will keep the home computer constantly aware of their location and physical condition. Speaker/phone jewelry worn by the elderly will allow them to converse with the computer --as well as make or receive a phone call --at any time.[15]

The home and family surely will be affected by new developments in computer education. It is certain that a stronger home-school link will develop as education expands from schools into the home, where adults can participate in learning activities such as computer homework and involve themselves in self-instruction.

As computers more and more take over those subjects and materials considered to be basic education, some writers suggest that schools, as well as homes, will be called upon to "teach values, motivation, and even religion.[16]

Networking, or linkage of a number of computers, appears to be the wave of the future. For example, in remote areas, like Alaska, the computer will become a

---

11. Francis, Jack, Appliance Fantasizing for the Decade and Beyond, Slide Set, "Patterns in Progress," Association for Home Appliance Manufacturers Meeting, Minneapolis, Minnesota, November 1-3, 1982, p. 9, 17, 19, 21.
12. Forum, J.C. Penney Company Inc., P. O. Box 2811, New York, 10001, Jan., 1982, p. 24, 25.
13. Naisbitt, op. cit., p. 46.
14. Compunications, TAP, 21 (Trends Analysis Program), American Council of Life Insurance, 1850 K Street, N.W., Washington, D.C., 20006, Summer, 1982, p. 7.
15. Mason, Roy and Lane Jennings, "A Trip to Grandmother's House," Popular Computing, Vol. 3, No. 11, Sept., 1984, p. 82, 83.
16. Naisbitt, Op. cit. p. 47.

valuable educational tool in widening horizons of learners through computer networks linked to learning centers.

Although the concept of the electronic cottage with its work-at-home approach is being accepted by our society, there will still be for some a need to select a career in the outside world of work. Family members will need an awareness of career opportunities. It is common knowledge that familiarity with computers is a strong vocational asset. According to industry specialists, computer occupations--such as programming and systems analysis--are expected to be the most rapidly growing in the nation over the next decade. By one estimate, 75% of all jobs by 1985 will involve computers in some way, and people who don't know how to use them will be at a disadvantage.[17]

One advantage of working in computer technology is that workers appreciate being treated as individuals, and the technology of the computer allows for individually tailored arrangements with each of thousands of employees.[18]

Trends

A present development is the use of telecommunication satellites, earth-orbiting structures which transmit audio, video, and data to and from earth stations. This satellite technology is the beginning of computer networking on a global scale. A vision of the future for the United States is that satellites will sweep back and forth across the country picking up and dropping large packets of information in each major city.[19]

Fiber optics, a communications technique using thin strands of special glass through which information travels on a beam of light, accelerates the speed of information transfer beyond that of metal wires. By 1997, the combined technologies of computers, telecommunications, and optical fibers will enable consumers to access more quickly many services such as shopping, banking, and information retrieval without having to leave home.[20]

A sign of future wide-spread computer knowledge can be found in the rapidly increasing number of summer computer camps being offered to young children between the ages of 8 and 17. Indications are that future generations will be computer wise.

Since the microcomputers are at the center of a rapidly growing and dynamic development, almost daily several new trends appear in the news. One such trend can be seen in classified advertising, where over the year of 1983 the number of classified ads offering "previously owned" computers had increased five times. Also there is a new publication, Computer Classified Bluebook, modeled after the Bluebook of the automobile industry and compiled by a group of Nevada businessmen, to be updated four times a year.[21] Other ads offer trade-ins on computers and opportunities to upgrade computers.

The technology for expansion of all these future trends is on the scene today, but much of it must be combined into a workable whole. As was the case with earlier times of change like the great Industrial Revolution, it is expected that changes will occur, but no one can accurately predict how far-reaching these changes will be in a process that is creative and self-developing but is also dependent upon a number of other factors in our society.

Where is the computer revolution leading us? It probably will take us into a complex combination of both good and evil. In the computerized society of the year 2000 and beyond, the moral choices will have to be made by our own good intentions even though we are not perfect.

Summary

Since microcomputers are having and will continue to have a significant impact upon both the home and the workplace, it is becoming increasingly imperative that those involved in any of the home economics roles acquire the knowledge and skills necessary for effective participation in the present Information Revolution. Two specific challenges facing home economists now are: (1) To develop software in child growth and development, family stress, parent education, clothing and textiles, housing and home furnishings, energy and equipment, and career education, and (2) To use software for record keeping and administrative purposes. Only when these efforts are made will home economists meet the challenge of helping to shape the future.

---

17. Naisbitt, op. cit., p. 33.
18. Naisbitt, op. cit., p. 43.
19. Lias, op. cit., p. 181.
20. Forum, 1982, op. cit., p. 5 and 6.
21. Family Weekly, 515 Broadway, New York, 10036, June 3, 1984, p. 14.

BIBLIOGRAPHY

CNN News, June 14, 1984.

Compunications, TAP, 21 (Trends Analysis Program), American Council of Life Insurance, 1850 K Street, N.W., Washington, D.C. 20006, Summer, 1982.

Family Weekly, 515 Broadway, New York, New York 10036, June 3, 1984.

Forum, J. C. Penney Company, Inc., Consumer Education Services, P. O. Box 2811, New York, New York 10001, January, 1982.

Francis, Jack, Appliance Fantasizing for the Decade and Beyond, Slide Set, "Patterns in Progress," Association for Home Appliance Manufacturers Meeting, Minneapolis, Minnesota, November 1-3, 1982.

Lias, Edward J., Future Mind, Little Brown, and Company, Boston, 1982.

Mason, Roy and Lane Jennings, "A Trip to Grandmother's House," Popular Computing, Vol. 3, No. 11, Sept., 1984.

Naisbitt, John, Megatrends, Warner Books, Inc., New York, 1982.

Rock Springs Daily Rocket-Miner, June 14, 1984 and July 5, 1984.

Spinrad, Norman, "Home Computer Technology in the 21st Century," Popular Computing, Vol. 3, No. 11, Sept., 1984.

Toffler, Alvin, The Third Wave, Bantam Books, March, 1980.

CAI AS PART OF THE
TOTAL FACULTY DEVELOPMENT PACKAGE

Kathleen Bellinger, Ed.D.

University of Louisville,Louisville, Kentucky

ABSTRACT

Computers are part of the top priority objectives for the next five years at the University of Louisville, like many universities and colleges where the strategic planning process has been used recently. The University-Wide Directions accept the assumptions that the application of computers is essential to virtually all University activity, all plans will take computer applications into account, and the use of current and appropriate computer resources enhances instruction, research and service and increases administrative productivity. This presentation describes the starategies and methods used by the Center for Faculty and Staff Development to integrate knowledge and use of computers into the total faculty development package. The Office of Computer and Information Services works with the Center. Computer literacy for faculty was a first step. Efforts to expand innovative educational delivery systems that are effective, including CAI, and developing pilot projects using advanced instructional technology in degree programs are summarized.

CAI AS PART OF THE TOTAL FACULTY DEVELOPMENT PACKAGE

Introduction

Rapid dramatic changes are occurring in our technologic information society. Generation and transmission of "knowledge" is the business of the universities. Faculty and students at colleges and universities throughout the United States demand computer resources. The University of Louisville is involved in this trend. Faculty view the computer as a tool to aid the teaching-learning process and to enhance scholarly productivity. Students view knowledge about computers and experience with the technology as prerequisite for career success and advancement.

University-wide Plan

A comprehensive plan is required to meet these needs. The plan for the University of Louisville identifies and addresses the unique needs of various academic units in the University, yet puts those needs into the larger context of the total institution. The interconnection of computer hardware including micros, minis and mainframes into a university-wide system --network--promotes the human networking of ideas and information necessary to meet both unit and university needs. The University expects faculty, administrators and students to locate, access and manage information without concern for technical details of interconnection to the network. Their concern should be obtaining, analyzing and synthesizing information. Movement towards implementing the University's strategy includes: establishment of a "high speed interactive network" for computing; enhancement of the University's dial-up facilities to allow faculty and students access to the University's computing network with time and location their choice; addition of microcomputer labs for instruction; development of microcomputer clusters for faculty and curricular development within each college; construction of several computer equipped classrooms to provide the tools to teach with computers; location of user centers at convenient places for computer services accessible to all users; and development of a "recommended" equipment list to guide purchase of hardware and software, thereby promoting the "connectability" of the equipment to the University's network. The approach proposed by the Ohio Board of Regents "Computer Task Force Report"(October, 1983) was used as a basic frame for the plan. Computer resources are assigned on the basis of the centrality of computing to an academic area. Fields such as computer science or engineering, in which computing is more integral to the curricular and research needs, require greater numbers of computers and more sophistication than disciplines such as music or philosophy.

Four "strategy levels of computing" tie different curricular goals with varying amounts of computer resources. The levels are:

Level 1 (Elective)--students may elect to take one or more courses in computing
Level 2 (Literacy)--computing is a basic area of knowledge which every student should understand
Level 3 (Skill) --computing is a universal skill which every student will use throughout their undergraduate program
Level 4 (Foundation)--computing is the key technology for information processing and will be used in all aspects of an academic program (University of Louisville, 1985a).

As the descriptions show, the four levels are based on the centrality of computing to the curriculum of the academic discipline and therefore determine the number and type of workstations allocated to each unit. The plan to create the baseline high speed interactive environment requires network and mainframe capacity with sufficient size and complexity to support a large number of workstations. Planning for the mainframe capacity and network has been underway since the previous academic year.

Implementation Strategies

The mainframe computer resources will provide computing capacity to support services commonly needed by many colleges and departments. A variety of high speed workstations for faculty and students will be connected to the network by hard wire and dial-up facilities, and the network will provide access to a mainframe computer. Megalink services will be provided between the mainframe computer resources and the Belknap, Health Sciences, Shelby, and Garden Court campuses. Optical fiber will support communication between buildings within a campus. The combination of these technologies should provide sufficient transmission rates to support the high speed interactive environment. Approximately 300 workstations will be connected to the network during each of the next five years. These resources will then support instructional computing and large scale research projected for a number of years.

Microcomputer labs, such as the Decision Support Lab in the School of Business building, will provide access for students to a broad range of education software. This lab contains 25-30 microcomputers and will provide access to software and peripheral devices such as high quality printers and a graphic plotter. Groups of workstations called "Curriculum Development Clusters" provide a convenient location in each college for faculty to evaluate and incorporate use of appropriate software into their teaching of specific courses.

To provide students with the opportunity to use computing resources in the classroom environment, two types of computer equipped classrooms are proposed. There will be a few other classroom labs like the Decision Support Lab to provide students with hands-on experience with computers under the supervision of faculty. There will also be classrooms equipped with easily visible monitors that will contain a workstation at which faculty can demonstrate either microcomputer or mainframe computer resources. These "wired classrooms" are being placed on each campus. Also, many faculty will benefit from workstations on their desks where they can access mainframe computing resources as well as professional office support functions such as electronic mail, scheduling, and text processing.

To implement the University-wide Strategic Directions (1985b), recommendations for a series of related academic policies and programs have been made to ensure ongoing support and direction for academic units as they implement information management concepts and tools into teaching and research. The recommendations include: 1) expanding the faculty development program related to information management and tools; 2) designing short and long term plans for curriculum development and modification, including faculty incentives; 3) negotiating discount computer purchase agreements for faculty, students and staff; 4) continuing the no interest loan program for faculty to purchase and use computers in teaching and research; and 5) negotiating an institutional license for microcomputer software, thus using the University's volume of purchases to save money.

Faculty Development

Many faculty do no yet have the computer skills necessary for the University plan to succeed. Each semester the mini-grant for faculty instructional development program has funded increasing numbers of computer related projects. Funding has been made available to support courses for faculty, for purchase of specific software and hardware needed by faculty to develop or implement computer-based instruction, for services of consultants, and for travel money for faculty to attend or present at computer related teaching or research conferences. Faculty development courses offered by the Provost's Office through the Center for Faculty and Staff Development, as well as short courses offered by the Office of Computing and Information Services continue to generate high enrollments and interest among faculty. More advanced and/or specialized courses are being developed and offered. These courses focus on specific computer applications e.g. advanced SAS or SPSS, authoring computer-based instruction, or learning a specific micro package such as dBaseII. Knowledgeable faculty are preferable as instructors; the issues of release time and course overloads are barriers that are being addressed.

The University is trying to discover and develop alternative ways to deliver this necessary instruction which requires less extensive and therefore expensive use of faculty time. Computer-based packages currently exist for many microcomputers that may be used for individual self-instruction. Similar materials for mainframe applications are less available, but the components to create them i.e., courseware development packages exist, and could be used to provide excellent instructional offerings.

Summary

Although the total integration of computing represents a legitimate and strategic response in an information society, requiring every faculty and student to own and use a personal computer for example, is not viable for most state supported colleges and universities. Excellence of programs must be carefully and continually balanced with fiscal responsibility. The University of Louisville is implementing a plan which is designed to accomplish this. (Specific examples of such programs follow in the oral presentation.)

References

The Ohio Board of Regents. (1983, October). Computer task force report. Columbus, Ohio.

University of Louisville. (1985a). Academic computing strategies and tactics : A strategic planning document. Louisville, Kentucky: Computing and Information Services.

University of Louisville. (1985b). University-wide strategic directions 1986-1991. Louisville, Kentucky.

The use of a graphic scratchpad for students in ICAI

Dick J. Bierman and Anjo A. Anjewierden

Psychological Laboratory
University of Amsterdam
Weesperplein 8, 1018 XA
Amsterdam, The Netherlands

ABSTRACT

A computerized scratch pad facility is described which enables the student to make drawings and notes while solving problems presented by a computer coach. Parsing the contents of the scratch pad yields information by which the computer coach can infer students cognitive actions, which are then compared with the computerized expert's actions. This comparision forms the basis for the tutorial diagnosis.

Introduction

A knowledge based computer program is under development that coaches students while they try to solve problems in the domain of physics. The system consists of four cooperating expert systems: the domain problem solving expert, the diagnosis expert, the tutor expert and the interaction expert. The domain expert solves the problem and the diagnosis expert compares the resulting trace with the actions of the student. On the basis of a diagnosis that follows from this comparison and from the student model, the tutor expert chooses an action according to an explicitly represented teaching strategy. For instance:

IF
   the diagnosis expert establishes that:
   1. the qualitative evaluation of the problem by the student is not OK with a probability larger than 0.80
   2. there is confusion about units with a probability larger than 0.50 and the studentmodel tells that
   3. the tranformation rules for units are not mastered

THEN
   the tutor experts chooses an intervention action according to the script about units.

Since the domain experts trace is supposed to reflect the cognitive processes of an expert, a comparison can only be made if we can obtain enough information from the student to infer his cognitive processes from his actions. To allow for this type of inference, the interaction between the computer coach and the student was initially based on menu's. These menu's were constructed from the domain experts knowledge base in such a way that each entry had a one-to-one relation with a cognitive process. However this approach had two major drawbacks:

1. The menu's contained only legal possibilities which prevented the elicitation of possible student's misconceptions.
2. All the inititiative came from the computercoach.

It has been found that the sketches of the student, while analyzing the problem situation, seem to be an important aspect of the problem solving behaviour and the information hidden in these drawings is relevant for the teaching strategy of the human coach.

BEST COPY AVAILABLE

In this paper we will describe a new graphic input system for the student. This input system is called the 'scratch pad' and it:

A: Significantly broadens the bandwidth of the information channel while it does not suffer from the two drawbacks associated with the menu approach.
B: Provides this information to the computer coach modules in an intermediate format that allows the diagnosis expert to infer the cognitive processes of the student in such a way that the tutor expert can decide upon tutorial actions.
C: Is imbedded in a general environment that is natural and suitable to stimulate student's initiative.

Implementation of the scratch pad

The graphic scratch pad has been implemented in PCE-Prolog. PCE-Prolog consists of the C-Prolog language and a user interface based on bitmapped displays with a mouse, and a window system (PCE). There are two kinds of windows, "pictures" and "views". Views can only contain text, while a picture can also display graphical information. One of the pictures is the graphic scratch pad which allows the user (student) to manipulate figures by means of the mouse. The available figures (which differ from domain to domain) are initialized through a few predicates added to C-Prolog. It is only when using these predicates that the programmer needs to bother with (relative) coordinates. (Parts of) texts may be 'picked up' by the student from other windows, e.g. the window with the problem text, and 'pasted' onto the scratch pad.

One of the important objectives of the described facility is to enable the evaluation of the drawing that the student makes of the problem situation in 'meaningful' terms. Thus it is essential that the representation of the scratch pad contents is such that the diagnosis expert within Prolog can draw conclusions in terms of cognitive processes. For instance the system must be capable to infer from the drawing that the qualitative evaluation was (or wasn't) correct (e.g. the asked volume is sketched larger, smaller or identical to the original volume).

So, if the drawing contains two separate composite figures, one representing the initial state, one representing the final state, the system must be able to identify both states and their differences.

The detection of distinct composite 'objects', such as the sketches which the student makes, within a picture is a notoriously difficult problem, let alone the classification and interpretation of these objects. However when dealing with a limited domain, like thermodynamics, only a limited number of primitive graphic data structures are sufficient to compose the sketches which represent problem situations. Thus the problem of recognition and classification is greatly reduced. Further reduction is achieved by limiting the number of ways in which the figures might combine.

The 'handle' concept

When the author of the computer coach defines the primitive graphic data structures as named prototypes like 'piston', 'container' or 'text', their possible relations with other prototypes are defined too. This is accomplished by indicating points on the figure called 'handles' (see fig. 1), a handle should to be thought of as a 'potential relation' between two prototypes.

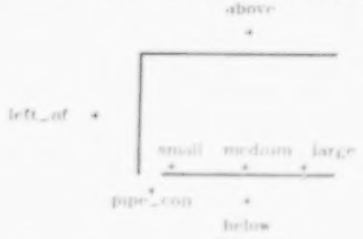


Fig.1
Example of the open_container prototype with some 'named' handles.

For each 'handle' a list of prototypes is given, that are allowed to link to this point.
(e.g. handleproto(medium, open_container, piston, 20, 5) etc.)
Anytime a figure (an instance of a prototype) is 'pasted' onto the scratch pad, the system scans around this figure for 'handles' of an already present (composite) figure to link to. If a handle is found the new figure is 'sucked' into the already present figure. (Unintentional links are easily removed by 'picking up' the last entry and dropping it outside the 'scratch pad').

BEST COPY AVAILABLE

PCE has its own database in which all information concerning the windows is kept, including the relations between the primitive figures of the composite figure. If no links can be made the figure is considered to be the start of a new composite figure.

The PCE database can not be accessed from Prolog but whenever from the Prolog environment the predicate "describe_scratchpad" is called, the PCE database is dumped in Prolog format into the Prolog database. An example of a problem sketch with the resulting Prolog clauses is given in fig. 2.

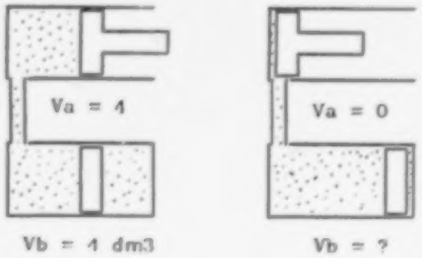


Fig. 2a
Problem sketch

```
relation(medium,   o_cont1, piston1).
relation(below,    o_cont1, 'Va=4').
relation(pipe_con,o_cont1, pipe1).
relation(pipe_con,c_cont1, pipe1).
relation(medium,   c_cont1, free_piston1).
relation(below,    c_cont1, 'Vb= 4 dm3').

relation(small,    o_cont2, piston2).
relation(below,    o_cont2, 'Va= 0').
etc. etc.
```

Fig. 2b
Resulting Prolog clauses

How the computer coach uses PCE-Prolog

The major advantage of the graphic scratch pad is that it allows the computer coach to infer student's reasoning without intervention. For example Prolog predicates are implemented like:

qualit_eval()

and

confus_unit()

which, given the database of fig.2 result in the answers that the qualitative evaluation is OK with a large probability and that there is some indication of a confusion about the units. To infer the latter a simple parser of textual information on the scratch pad was written. This parser parses the clauses given in fig.2 into a trace that is comparable with the expert problem solver's trace. For instance the parser parses the second clause in fig.2 into

```
pupil_step(fill,[givens,[volume,4]]).
```

while the trace produced by the domain expert has the following clause:

```
expert_step(fill,[givens,[volume,4,dm3]]).
```

Thus the diagnosis expert has the necessary information in the required format to compare student and expert and to conclude that there might be confusion about the units.

However, at present it is practically impossible to analyze the scratch pad any time a simple figure or text is added. Thus, the system is not capable to intervene if one wished to monitor for errors made while the student is making the sketch. We decided to call the 'describe_scratchpad' predicate whenever the student chooses an action that implies he has left or interrupted the analyzing stage of the problem solving process (i.e. when he starts to look up the formula window, the constants window, the calculator window or when he starts 'talking' with the computer coach). Since the computer coach is a.o. intended for research in teaching strategies it is eventually necessary to monitor the development of the drawing in order to switch to a direct intervention strategy. Apart from this limitation the PCE package as a whole is certainly a useful tool in eliciting student's initiative in a form that is interpretable. For instance, we chose the "view" with the problem text to be one of a series of overlapping windows. If one of the other windows is activated, for example to inspect and copy a formula, the problem text disappears. This stimulates the student to copy relevant parts of the problem text to the scratch pad. Thus it is easy, using the text parser, to diagnose without explicit questioning if the student failed to recognize a relevant item in the problem text. This use of PCE might be compared with the use of a eye-movement recorder. Another example is the formula "view", we limited this "view" in such a way that only a single formula at a time can be inspected

BEST COPY AVAILABLE

by the student. So we can evaluate if and how long the student focusses on a formula.

Conclusions

Preliminary experiments with a version of the package (outside of the Prolog environment) in which the role of the computer coach was played by a human coach at another terminal, showed that the intervention rate decreased with about 50% due to the scratch pad facility.

We feel that the graphic scratch pad and the overall environment of the coach provides an interesting and novel approach to the elicitation of knowledge from the student. Multiple choice menu's give away cues that the student should discover by himself while free response parsing is a huge load for the system and often results in 'undecided' conclusions. The computer coach with the scratch pad avoids the drawbacks of these extremes.

The handle concept seems a powerful and general method for keeping the complex and error prone user interaction within limits. Parsing composite figures connected by handles turns out to be rather easy if the number of figures is not too large.

Acknowledgement

The work described in this paper was partly funded by SVO (Foundation for Educational Research).

BEST COPY AVAILABLE

Factors of Flexible CMI System Design

Robert L. Jorczak

Control Data Corporation

ABSTRACT

The design of CMI systems is a complex task envolving the specification of system functions and other considerations. Designing a generalized CMI system that can meet the management needs of a wide range of users is even more difficult as system flexibility and functional scope are additional issues. This paper surveys potential CMI design issues giving special emphasis to flexibility of system functions.

PURPOSE

The purpose of this document is to identify the key design factors of Computer Managed Instructional (CMI) systems intended to provide comprehensive and flexible capabilities to system users.

CMI systems that will support a small, fixed range of capabilities, and be delivered in a fixed software and hardware environment, are relatively easy to design. A CMI system intended to meet the needs of a large and diverse group of users, however, must consider a wide range of functional and delivery requirements.

This paper identifies functional and other design parameters that must be considered in designing any CMI system, but special emphasis is placed on maximizing system flexiblity so that the system can meet the differing requirements of system users.

This paper first identifies CMI goals and then discusses some issues of overall CMI system design. It proceeds to discuss important factors of major CMI authoring and delivery functions.

CMI DEFINITION

Computer-Based Education (CBE) encompasses all aspects of using computers as part of an educational process. This includes primary delivery and control of instruction, Computer Assisted Instruction (CAI), and computer aided support functions such as Computer Managed Instruction (CMI).

CMI is the use of a computer to manage both the delivery of instructional materials (including CAI materials) to students and the maintenance of data and records that result from the educational process. Obviously, these functions are closely related. Decisions about when, what, and how material should be delivered to the student must be based on the historical and progress data available to the system.

Management of instruction has always concerned two issues: accountability and efficiency.

Accountability can be summarized by a simple question: Does the instructional system work? The key issue of instructional systems always concerns the ability of the system to enable students to achieve instructional goals.

Efficiency issues involve practical matters such as resource allocation, cost of operation, and average time for students to complete curricula.

Modern educational theory has added at least one more concern to the management of instruction: individualization (Allen, 1975). Individualization is related to both accountability and efficiency. The recognition that each student requires unique management of instruction for optimal educational progress has greatly complicated learning management.

The goal of providing effective, efficient individualized instruction places a huge burden on instructors and instructional systems. Teachers and trainers typically deal with hundreds of students. Attempts to manage and record the progress of each student tax the managing and accounting capabilities of even the most diligent instructors.

It may be true that using a computer to manage instruction is the only way individualization of instruction can become cost efficient and, therefore, commonly available.

SYSTEM DESIGN GOALS

Given that the goal of any instructional management system is to make the instructional process accountable, efficient, and individualized, what factors are important in the design of CMI systems?

A comprehensive CMI system attempts to provide system users, including authors, instructors, administrators, evaluators, and students, with utilitities for accomplishing instructional management. Issues and features required to accomplish this task are considered in the following sections.

CMI system design, however, should not be confined to only management functions. System designers must consider the capabilities of the computing environment. A system designed without regard for the computing environment may not perform adequately in terms of speed or memory requirements.

Another design consideration is system flexibility. Management functions must be flexible to meet the needs and preferences of users. Another aspect of flexibility is system modularity. A highly modular system is divided into self-contained modules that communicate with each other in well-defined ways. Further, each module can be replaced by programs and applications that are not part of the system if such applications follow established communication formats. A system designed to be highly modular has greater utility as users do not have to use the whole system, and can substitute customized modules for standard system modules.

To summarize, the design goals of a general CMI authoring system include:

- o Comprehensive and Flexible Management Functions
- o Adequate System Performance within Computing Environment
- o A High Degree of Modularity of System Components

AUTHORING FUNCTIONS

It is convenient to separate CMI system capabilities into two groups: authoring functions and delivery functions. Authoring functions are required to build CMI curricula that include components such as objectives, tests, and test items. CMI delivery entails all functions required to enroll students in curricula and monitor student progress.

Major CMI authoring functions include:

- o Organization of Curricula and Courses
- o Assessment of Need and Achievement
- o Prescription to Remedy Needs
- o Definition and Storage of Student Performance Data and System Effectiveness Data

These functions define major system components that can usefully be made modular. Authors may then choose which functions are required to meet their needs and which functions may be handled by utilities not part of the complete CMI system.

Organization

A major authoring function a CMI system should provide is the organization of instructional units. The system should allow curriculum authors to organize assessment and instructional components into coherent structures. Such structures determine the instructional domain available to the learner including what topics can potentially be assessed and/or taught.

For authors to define and organize instructional units, they must be able to specify the following:

- o Unit Components
- o Component Presentation Strategy
- o Unit Scoring Procedure

A flexible CMI system will allow authors to include different components in the CMI unit. Such components should include:

- o Learning Objectives
- o Assessment Tests
- o Learning Resources
- o Other Software Applications

Hierarchical structures are very useful in organizing CMI units. To achieve such hierarchies, the system should allow CMI units to be included as components of a CMI unit.

After components comprising the CMI unit are specified, the author needs to determine how the units will be presented to students. Simple algorithms, such as sequential and random presentation, are useful. Perhaps most useful is presentation via a student selection menu, where components are offered for selection by each student. A general CMI system should make it easy for an author to achieve any of these simple routing alternatives.

A comprehensive system should allow more flexible and complicated routing. Authors may wish to conditionally route certain students to certain components or alternately present unit components based on performance or other data.

CMI authors should be able to establish a unit scoring procedure based on the scores reported from components. Scoring procedures can be relatively simple, such as unit mastery based on completion or mastery flags reported from components. Examples of other scoring procedures include unit score based on:

- o The Score of a Single Component
- o The Summed Scores of Components
- o A Number or Percentage of Mastered or Completed Components

Assessment

Until the advent of effective artificial intelligence in the field of computer-based education, assessment means testing. It is possible to assess needs based on human-supplied information of a student's history, or other information, but testing is the heart of most CMI assessment.

Testing involves two general areas: the construction of test items and the construction of testing strategies that control the delivery and scoring of test items.

Some CMI systems have test building interfaces highly integrated into the design of the system. This tends to tie the management system to the capabilities and limitations of the testing component. This can be useful for users looking for complete packages without the complication of alternative testing strategies, but it is not desirable from a flexibility viewpoint as many potential customers already have testing vehicles and/or may wish to use different testing strategies.

Test and test item construction are functional areas that should be modular in a flexible CMI system. The testing utility ideally should allow tests to be constructed of test items built with alternate test item editors. The CMI unit should accept information from tests of various testing vehicles (including tests that are not computer-based) that report test data in the format established by the CMI system.

Test Item Construction

There is no functional difference in test questions used for CMI assessment and those used in instruction except, perhaps, in feedback and repeated trial requirements. This means that computer-based test item building utilities can serve both CMI tests and CAI lessons. A flexible CMI system would be able to use test items generated with different software utilities as system users may have large test item databases already in use on the computing system.

CMI test items almost always involve selected response and simple short-answer questions. Subjective essay and partial credit type testing requires human judging capacity in most situations.

CMI authors frequently require sophisticated capabilities for authoring test items. This includes the capability to perform complex judging of student inputs. Complex feedback and item scoring capabilities may also be required by authors to achieve their assessment needs.

Test Construction

Factors important in test construction include specification of the following:

- o Test Objective
- o Potential Test Items
- o Test Item Selection and Presentation Algorithm
- o Test Scoring Algorithm

Most frequently, CMI assessment involves testing whether a student meets the criteria for a learning objective. This information is the basis for assignment of learning resources and continued progress through the CMI unit. A complete CMI system will allow a description of the objective to be included with other test specifications. This objective may be shown to students before, during, or after testing at the discretion of the author.

Of course, some testing may not have specific learning objectives but may wish to measure student performance based on a reference norm of all students taking the test. A flexible system will also support this type of testing.

Besides optionally allowing an author to specify testing objectives, a test construction utility must allow the specification of all test items that may be used in a test. Again, the most flexible systems will accept test items from several test-item-building utilities.

Testing effectiveness is related to the range of test item presentation procedures offered by the test building utility. Standard presentation algorithms include sequential and random item presentation. Often authors wish to present from groups of associated items to ensure that students are presented items of certain types, items covering certain topics, items of a particular difficulty level, etc.

More advanced item presentation algorithms conditionally present particular items based on student test performance or performance on each test item such as adaptive or tailored testing (Stocking, 1982). A comprehensive test construction utility should support such testing strategies.

Another aspect of item presentation is test termination. More efficient testing can be accomplished if authors are given flexible options for terminating the test. For example, an author may wish to terminate a test as soon as the student fulfills the mastery (scoring) criteria or as soon as it becomes impossible to meet the mastery criteria. Some authors may wish to conditionally vary the number of items in a test based on student performance (Heines, 1980).

Test scoring options is another area where authors may require flexibility. Author-determined scoring may be simple mastery based on the number of correct items. Often authors want a numerical score based on the number of test items correct or on points amassed from all items presented.

Essential Testing Data

If the assessment function is to be truly modular, all significant test data must be passed to and from the assessment module to the CMI unit which then adds the information to the student's performance data. It is important that the communicated data be sufficient to meet the needs of the system and the unit author. This testing information should potentially include the following:

1. Test Objective(s)

This is a precise statement of what is being assessed by the test. In many academic and training situations this would be a learning objective or behavioral objective.

2. Testing Status

This is information about the delivery status of the test. The possible status states include: not taken, in progress, completed, exempted, not available, etc.

3. Test Results

This is information about the student's performance on the test. It may include global test information such as a score for the whole test. It may also include detailed information such as a score on sub-sections or objectives of the test. A versatile CMI system should be able to accept general test results and accept more detailed information if it is provided in the proper format.

Assignment

Once an instructional management system assesses the capabilities and needs of a student, it proceeds to prescribe actions and recommend materials that will enable the student to achieve educational or training goals.

The basic requirement can be simply stated: Given an assessment that indicates skills and knowledge that are present or lacking in a student, find (through some algorithm) a set of learning resources that addresses these skills and knowledge. More complicated algorithms may be desired: find the smallest set, find a set that best excludes the skill or knowledge that is present. A versatile CMI system should offer various assignment options.

A flexible CMI system does not restrict the definition of the learning objectives (skills, knowledge, etc.) or the types of learning resources it can assign.

Advanced CMI systems may provide aid in matching learning resource types with particular student characteristics. Advanced systems may also consider student-specific data in making an assignment, such as historical data or learning style. Ideally, the system would allow authors to determine the assignment algorithm including its input and output.

In designing a CMI system, it is important to understand that the output of the assessment (testing) system is the input of the assignment system. Therefore, the learning resource assignment module has as its input the essential test data previously described.

Any of this information may affect the learning assignment. For example, the testing strategy may be required or useful in determining what should be assigned. Obviously, the most important information is the scoring procedure and competency scores established for a specific test.

This information is not sufficient for the learning assignment utility to accomplish its task. It must also know just what skill or knowledge was assessed, and what score is considered indicative of possession of a skill. It is, therefore, necessary that the assignment module store or have access to information about available learning resources including a description of the educational objectives addressed by each resource.

Stored Data

Another major CMI authoring function is the definition of data to be stored by the system. This data may be used by the system or may be included in reports produced by instructors, administrators, or evaluators. CMI authors must define which data to store for monitoring and reporting student status; and analysis of test items, tests, and learning resources.

Student Progress Data

A CMI system must store student performance information. Such information includes the testing data for tests already taken by a student (or input from another source). This information is necessary for the system to decide just where the student is in a particular CMI unit. Performance data generated by the system should be used by the system in determining which learning resource to assign.

Performance data is also required by administrators, instructors, managers, etc., who monitor student progress. Students also need to see information about the objectives they have accomplished and the progress they have made.

Certain performance data is defined in organizing CMI units. When authors specify scoring procedures and student access for a unit and its components, the system will automatically define the necessary variables and data records for this information. Authors may, however, wish to further define data to be available for progress reports. A comprehensive system will allow authors to define such data.

The reporting of performance data is another area where new potential for flexibility exists. Many existing CMI systems store data in a non-general data format and have data reporting utilities that present predefined data in limited ways. The performance data and format required by instructors varies greatly. A general CMI system should either make its data available to flexible, general purpose, report generators or provide a flexible one with the system.

Database management and report generation are not functions unique to a CMI system. System flexibility will be enhanced if authors, or other users, can specify an standard database format for their data. This would allow instructors to use a database manager they may already be using or to choose to obtain a database manager suited to their needs.

Analysis and Effectiveness Data

Effectiveness data record how effective a learning resource is at enabling a student to meet the mastery criterion (or passing grade) of the test for a particular objective of the assessment system. The CMI system should offer this capability or allow authors to define and store such data.

Another validation of interest is the effectiveness of the assessment vehicle. This can be accomplished via well-known testing and item analysis methods, some of which are already extant as computer applications. The CMI system should support the definition and storage of data required by such applications.

DELIVERY FUNCTIONS

The following tasks are typical for instructors and/or administrators in managing instruction.

- o Roster Students
- o Inspect/Report Student Progress
- o Perform Operations on Student Groups
- o Establish Access and Restrictions
- o Define User-Specific Student Data
- o Schedule CMI Resources

A comprehensive CMI system should provide help in accomplishing all of these tasks (Callahan, 1984). Many of these tasks can only be accomplished in a cost effective manner by using the capabilities of computers.

CMI system delivery is also concerned with the product format and delivery system hardware. What hardware is required to support CMI for students? Where are student records stored? Does each student have a flexible disk containing his or her records? How many students are supported by the system? How much courseware can be presented? If the system requires the use of several flexible disks, how do the disks work together? These real-world matters are sometimes overlooked in CMI system design.

Many potential CMI users are looking for a means to monitor the progress of students who are located at geographically diverse locations. This implies a system that can distribute CMI courseware to students but maintain performance records at one, central location.

The capabilities of modern micro computers allow system configurations where CMI courseware execution can take place in the terminal of the student of instructor. Courseware can be distributed to these delivery stations via flexible disks or communication links. Student records generated by the CMI system can then be sent to centrally located databases via communication links such as telephone lines. This record updating process can be accomplished automatically as the student proceeds through the curriculum if communication links are always present and information transfer fast enough. If such links are established only at certain times, administrators or the system can initiate the central records update from a record buffer containing the updated results of student performance.

CONCLUSION

The early development of Computer-Based Education was concerned with the authoring of instructional courseware. Many CBE developers are now turning their attention to the management of instruction. CMI holds great promise as data storage and manipulation has always been a strength of computing systems.

The capabilities of modern computers enable the development of generalized CMI systems that provide for highly individualized instruction tailored to the needs and preferences of a wide range of educators. Should CMI systems provide the features and flexibility needed in academic and training situations, we can expect that advanced instructional techniques will become widely available.

REFERENCES

Allen, M. W. "Computer Managed Instruction: A Definitive Design" Computers in Education, O. Lecharme and R. Lewis Eds. North-Holland Publishing Company, 1975.

Callahan, R. C. "A Redefinition of Integrated Computer-Managed Instruction". A paper presented at the EdCompCon Conference, November 1984.

Heines, J. M. "The Reliability of Sequential Testing" A paper presented at the Third Canadian Symposium on Instructional Technology, February 1980.

Stocking, Martha L. "Tailored Testing" A paper presented at AERA Presession on Item Response Theory, March, 1982.

A Microcomputer-based CAI Teaching Materials Production Support System

Masahiko KURATA, Hiroyasu CHIMURA, Yasuhiro MORIMOTO and Takahiro SATO

NEC Corporation

ABSTRACT

This paper describes two microcomputer-based CAI teaching materials production support systems. The construction procedure for instructional program in CAI is divided into the following phases : (1) Analyzing the instructional contents. (2) Constructing the structures of instructional elements. (3) Determining instructional sequences and designing teaching dialogues. (4) Writing teaching frames and instructional sequences on computers. One system described here helps teachers or subject matter experts to efficiently and systematically determine hierarchical structures of instructioanl materials in phase (2). The second system is a teaching materials authoring system which enables teachers and experts who are non-programmers to easily draw pictures and write text in phase (4).

1. Introduction

The construction procedure for instructional materials in CAI is divided into the following phases : (1) Analyzing the instructional contents. (2) Constructing the structures of instructional elements. (3) Determining instructional sequences and designing teaching dialogues. (4) Writing teaching frames and instructional sequences on computers.

Section 2 discusses a method for determining hierarchical networks of instructional elements and a computer-assisted support system for constructing the hierarchical network of instructional elements. Section 3 proposes an authoring system for writing teaching frames and instructional sequences on computers.

2. Cnstruction of hierarchical networks of instructional elements

This section describes a method and a system for constructing the hierarchical networks of instructional elements (Sato, 1978, 1979). It is necessary for classroom teachers and subject matter experts, when designing their own instruction or teaching programs in computer-assisted instruction, to clarify the instructional goals and contents of courses from the viewpoint of the instructional process.

The basic method is to first divide the terminal objectives into a number of elements (instructional units), determine the relationships between them, and thereafter visibly locating the elements on an easily interpreted hierarchical network graph with using graph theory and the computer system we will describe. The method discussed here is based upon the Interpretive Structural Modeling (ISM) method, which was originally developed within the field of social systems engineering (Warfield, 1976).

2.1 A method of constructing hierarchical networks of instructional elements

The process in determining hierarchical networks of instructional elements is shown in Figure 1.

1) Extraction of instructional elements

School teachers and subject matter experts divide the terminal objectives into instructional elements (instructional units). These elements are extracted from a textbook and teaching material, as the occasion calls for.

2) Determining relationships between all the elements

School teachers and subject matter experts determine the direct hierarchical relationships between the elements. The diret relationships are not only the formative relationships but also their directional relationships between superior and subordinate elements. They do not have to wory about the indirect or mediate prerequisites. They only have to decide what the immediate prerequistes are for each element. Their relationships between elements can be carried out from the viewpoint of the teacher's or the expert's logic as well as from the viewpoint of overall relationships. Decisions can be based upon experience-based instruction and judgment.

3) Structural relationships between all the items drawn as a digraph

After determining the subordinate objectives which directly underlie each instructional element, a computer is used to efficiently and automatically draws a hierarchical structure network graph, which includes all of the elements.

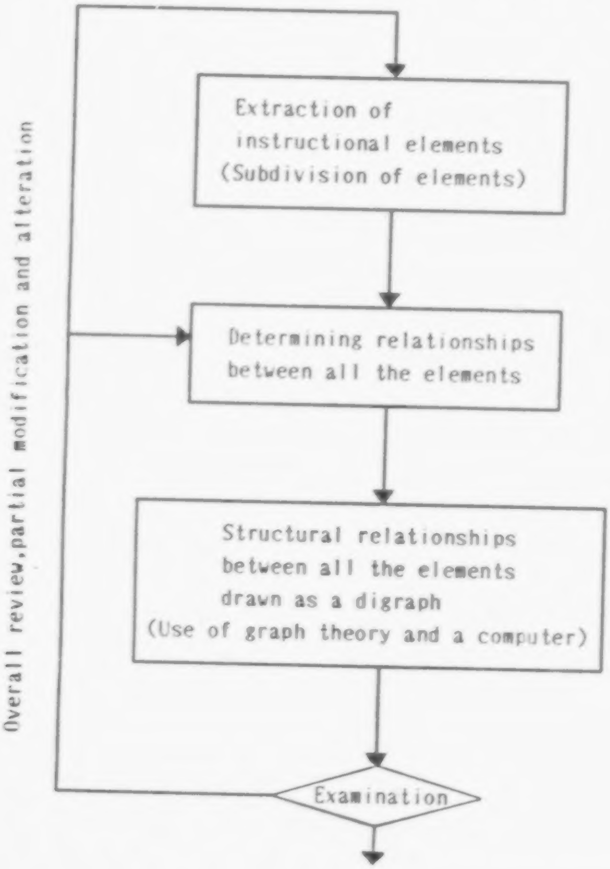

Extraction of instructional elements (Subdivision of elements)
Determining relationships between all the elements
Structural relationships between all the elements drawn as a digraph (Use of graph theory and a computer)
Examination
Overall review,partial modification and alteration

Fig.1 Process in determining hierarchical networks for instructional elements (instructional units)

4) Partial modification, alteration and over-all review

Partial modification and alternation in locating the relationships between elements can be carried out while observing the hierarchical relationship graph. Or, as necessary, overall review through modification of the subdividing for the elements can be conducted. Then, another hierarchical network graph is prepared.

The example in Figure 2 shows a hierachical structural network graph for teaching units of a lesson on "demonstration of the theorem : a straight line between the midpoints of two sides of a triangle is parallel to the remaining line and half the length of the remaining line." taken from a secondary school geometry lesson. In this figure, solid lines correspond to hierarchical networks of instructional elements.

2.2 Determination of instructional sequences

A strategy for determining the instructional sequence depending on the hierarchical network graph of instructional elements is described. While looking at the hierachical network graph of instructional elements, the instructional sequence can be determined from the pedagogical point of view. The first step is to detect the main cluster in the instructional sequence. Next is to detect the sub-clusters which are branches of the main cluster. It is not so difficult to detect some clusters, since the network graph of instructional elements is expressed as a hierarchical structural graph. The sequencing of elements in a cluster is determined, so that it flows from the subordinate elements to the superior elements as a general rule. In case of inductive studies, it may be appropriate to be determined so that the sequence flows from the superior elements to the subordinate element. Decisions of instructional sequences should be determined as a result of experience-based pedagogical points of view.

Next, the actual teaching dialogues are designed corresponding with instructional elements. An instructional element does not always correspond to a teaching dialogue unit (a teaching frame). It may be appropriate in some cases that a instructional element corresponds with plural teaching frames. In case of an example shown in Figure 2, two teaching frames (frame Q1 and frame Q2) are added in the sequence. After determining the instructional sequences, the next step is to design concrete teaching dialogues and an actual screen layout. They should be designed from th pedagogical point of view by classroom teachers or subject matter experts.

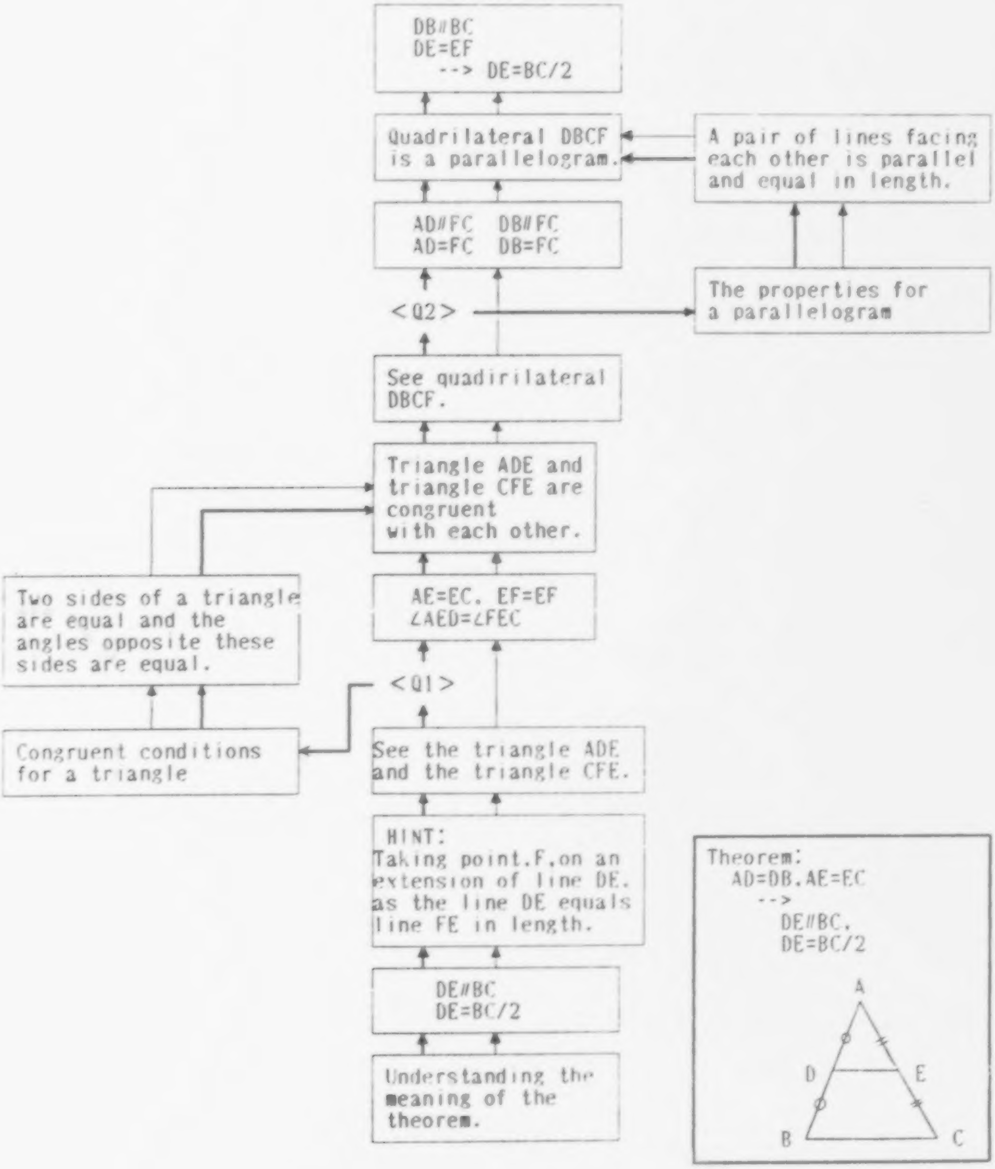

DB∥BC
DE=EF
--> DE=BC/2
Quadrilateral DBCF is a parallelogram.
A pair of lines facing each other is parallel and equal in length.
AD∥FC DB∥FC
AD=FC DB=FC
<Q2>
The properties for a parallelogram
See quadirilateral DBCF.
Triangle ADE and triangle CFE are congruent with each other.
Two sides of a triangle are equal and the angles opposite these sides are equal.
AE=EC, EF=EF
∠AED=∠FEC
<Q1>
Congruent conditions for a triangle
See the triangle ADE and the triangle CFE.
HINT:
Taking point.F,on an extension of line DE, as the line DE equals line FE in length.
DE∥BC
DE=BC/2
Understanding the meaning of the theorem.
Theorem:
AD=DB,AE=EC
-->
DE∥BC,
DE=BC/2
A
D
E
B
C

Fig.2 Hierarchical network of teaching elements on "demonstration of the theorem : a straight line between the midpoints of two sides of a triangle is parallel to the remaining line and half the length of the remaining line."

Solid lines correspond to hierarchical networks of teaching elements. Thick lines correspond to instructional sequences.

2.3 A computer-assisted support system for determining hierarchical structures of instructional material

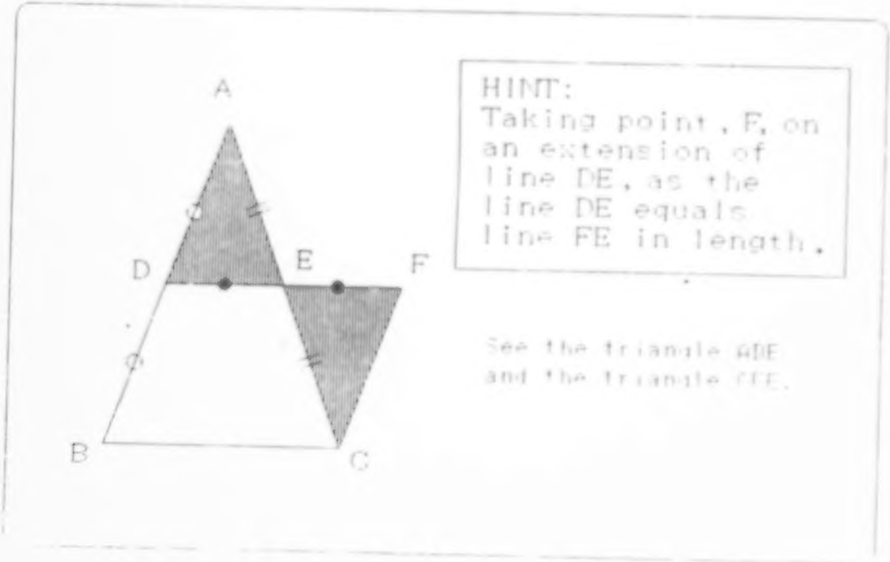
The authors developed a computer-assisted support system for determining the hierarchical structures of instructional material applying the method mentioned above (Chimura and Sato, 1983, 1984). The use of this system helps teachers or subject matter experts to efficiently and systematically determine the instructional structures.

The system has been installed on a microcomputer. Extracting the instructional elements, classroom teachers and subject matter expert store the elements in disk storage files (the element pool). A hierarchical network graph of the elements is automatically provided by computer after determining the immediate prerequisite(s) for each element. The elements are located where they can easily be seen on the interpretive network graph as shown in Figure 2. The set of elements and the set of network graphs can be stord in the disk storge files, and can be recalled to draw and revise them.

3. Writing teaching frames and instructional sequences

The section proposes a graphical teaching frame production system and an instructional sequence describing system depending on a new instructional sequence control method.

It is difficult to make an arrangement between subject matter experts and computer programmers. That is, it is difficult to correctly transmit the image the experts have in their mind to computer programmers. The classroom teachers and subject matter experts who are non-programmer are able to easily use the system described here without using any computer programming languages. This system is installed on a microcomputer. The operation for drawing picture on a computer screen is as easy as drawing pictures on a piece of paper with a pencil, a ruler and compasses. This system uses an electronic pressure sensitive tablet digitizer as an input device. Operating a stylus on an electronic pressure sensitive tablet digitizer is very close to writing on paper with a pen. Authors can trace a draft on the tablet digitizer. This system provides a specially designed sheet, which is set on the tablet digitizer. The menu of selected commands for writing pictures such as line, box and circle, and commands for editing pictures



HINT:
Taking point, F, on an extension of line DE, as the line DE equals line FE in length.

See the triangle ADE and the triangle CFE.

Fig.3 An example of an actual screen layout (a teaching frame) which is corresponding to the lesson shown in Figure 2. This is a scene of synchronized presentation of the figure and the text.

is printed on these sheets like icons. At the stage of creating graphic teaching-materials, the authors can draw figures, paint, write text and edit displayed pictures, interactively using pertinent commands on the sheet. Creating animation usually requires a lot of elaborate effort. The animation commands in this system enable authors to easily generate simple animation, such as the movement of an object and the flow of fluid. The author moves the stylus, keeping it in contact with the sheet to point at a specific command, and indicates a few points of which the figure is composed. In an advanced stage, the user can define his own commands, such as for drawing trigonometric curves, logarithmic curves and cycloid curves with writing programs in BASIC. These commands are registered and can be used in the same way as present commands. In Figure 3, a simple example of a teaching frame corresponding to the lesson in Figure 2 is shown which is drawn using this system.

Next, an instructional sequence describing system depending on a new instructional sequence control method is proposed. The classroom teachers, who are not familiar with computer programming, can describe an instructional sequencing (Sato and Chimura, 1984). The instructional sequences in Figure 4 can be expressed as a frame sequence table shown in Table 1. The elements on the row in the table correspond to teaching frame numbers and the elements on the column correspond to alternative key-sets. The numeral in the element of the table shows the turning teaching frame number. At the stage of the example shown in Figure 4, when the teaching frame number 1 has been displayed, the teaching frame number 2 and number 3 will be displayed by selecting alternative key number 1 and number 2 respectively. As the blank matrix is displayed on the computer display screen, the user only fills in the blanks in the form of the matrix with the turning teaching frame numbers.

4. Conclusion

Instructional materials development and production is a very important aspect in teaching. This paper described two microcomputer-based CAI teaching instructional materials production support systems. They are practical for use and have been used at some schools and educational centers in Japan.

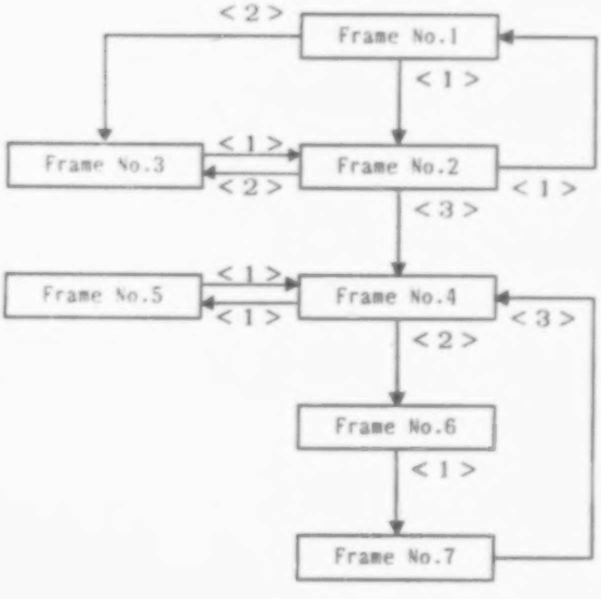


Fig.4 An example of teaching frame sequencing

Table 1 A key initiate instructional sequencing matrix
An author only fills in the blanks in the form with the turning fram number.

Frame \ Key	<1>	<2>	<3>	<4>
No. 1	2	3		
No. 2	1	3	4	
No. 3	2			
No. 4	5	6	7	
No. 5	4			
No. 6	7			
No. 7				

References

Chimura,H and Sato,T (1983) Support System for Producing ISM Instructional Units Network Using a Personal Computer. IECE Trans. Educational Technology, ET83-5, pp.39-42 (in Japanese).

Chimura,H. and Sato,T (1984) Computer-assisted analysis and determination of instructional sequences, Proceedings of the EdCompCon Conference, IEEE Computer Society

Sato,T. (1978) Hierarchical Display of Networks of Teaching Elements Using the Interpretive Structural modeling. IECE Trans. Educational Technology, ET 78-4, pp.23-28 (in Japanese).

Sato,T. (1979) Determination of Hierarchical Networks of Instructional Units Using the Interpretive Structural Modeling Method. Educational Technology Research, vol. 3, pp.67-75.

Sato,T. (1980) Data Analysis methods for Instructional Design and Evaluation. Meiji Tosho, Tokyo (in Japanese).

Sato,T. and Chimura,H. (1984) CAI Frame Sequence Control System. IECE Trans. Educational Technology, ET84-2, pp.35-40 (in Japanese).

Tatsuoka,M. (1978) Recent Psychometric Developments in Japan : Engineers Grapple with Educational Measurement Problems. Paper Presented at the Office of Naval Research Contractor's Meeting on Individualized Measurement, Columbia, Missouri, September 19, 1978, pp.1-19.

Warfield,J.N. (1976) Societal System. John Wiley & Sons, New York, pp.204-284.

TAIGA

A GRAPHIC AUTHORING SYSTEM

Eelco F. Laagland

Twente University of Technology, The Netherlands

ABSTRACT

In this paper we will describe the software system TAIGA (Twente Advanced Interactive Graphic Authoring System) which can be used as an aid to produce, distribute and evaluate courseware. Though TAIGA consists of several subsystems, this description will focus on the subsystem the author uses to implement courseware. This subsystem is named COPRO.

INTENDED USERS

TAIGA is intended to be used by courseware constructors, requiring no detailed knowledge of programming languages or computer (terminal) hardware. Though the ability to program in a high level language is not a prerequisite skill to use TAIGA, algorithmic thinking is desirable to a certain extent.

PHILOSOPHY BEHIND TAIGA

Construction of CAI is a complex activity. Educational goals have to be translated to a computer program running on a certain hardware system. This (layered) process requires various disciplines to cooperate, i.e. teaching, computer science, graphic and ergonomic design etc. TAIGA can be seen as an aid to structure this cooperation and decrease the time needed to implement various demands, by offering both a high level user interface to the inexperienced computer users and a low level interface to the (system) programmers. TAIGA consists of an integrated set of tools for each of the disciplines.

ERGONOMIC DESIGN CONSIDERATIONS

In order to appeal to a broad spectrum of users, most of whom have no detailed knowledge of computer hardware or software, special attention has been given to the ergonomic aspects of TAIGA.

TASK STRUCTURING

The task of developping instructional materials can be conceived as a layered process. TAIGA is designed to reflect this layered structure. Three layers (levels) are discriminated and the user is always informed of his 'location' in the structure. This prevents 'getting lost' and enforces the production of highly structured CBE materials.

INTERACTION

Interaction refers to the way information is presented and the way the user can 'command' the system. Two keywords apply here : viewports and functionkeys. In order to encourage structural encoding of the instructional content, the screen is divided into several so called viewports. Structural encoding refers to a technique whereby information is classified and each class of information is assigned a different part of the screen (port), e.g. a textport for orientating texts, a port for displaying questions, a feedback port etc. Most of these ports have default sizes and positions, but these can be overridden by entering user defined values. The only fixed ports are the status and command port for which the three top lines of the screen have been reserved. Depending on the situation some of these ports may overlap, e.g. the text and graphics port overlap in order to be able to mix textual and graphical information. The user can command TAIGA using function keys. Dedicated functionkeys initiate the commands with a single keypress. Some functionkeys have the same meaning regardless of the level TAIGA currently

BEST COPY AVAILABLE

functions at, other less frequently used keys have different effects on different levels of TAIGA.

THE SOFTWARE

TAIGA is written in PASCAL. There were to reasons for choosing PASCAL. First because this high level language offers an easy implementation of the required datastructures and second because it's widely available, not only on mainframes and minicomputers, but also on most microcomputers. The microcomputer version is written in MS-PASCAL and runs on MS-DOS machines for which a GSX-86 device driver is available. GSX 86 adds graphic capability and a device independent operating environment. This approach promotes the portability of TAIGA itself and of produced courseware.

THE HARDWARE

The pilot system was initially developed on the central computers of the TUT, a DEC 1080 and a DEC 2040 running TOPS 10. The terminal used was a DEC GIGI, a graphic color terminal. The unpredictable respons times and the high distribution costs on a timesharing system led to the desire to realise a microcomputer version of TAIGA (Micro TAIGA). Micro-TAIGA was released Oct, 1985. In order to promote portability Micro-TAIGA runs under MS-DOS (PC-DOS) and GSX-86. This way a wide range of microcomputers is supported. The main source of incompatibility is the implementation (and resolution) of graphics on the various 'IBM - compatibles'. This problem can be solved if the manufacturer supports GSX-86 (meaning a GSX-86 device driver is written for the specific graphic hardware).

COURSE STRUCTURE

A course developed with the aid of TAIGA has a hierarchical content structure. Each course consists of a maximum of 10 modules, each module contains at most 10 paragraphs and each paragraph no more then 10 episodes. Each level serves a specific function (level 1: lesson content structuring, level 2: instructional interaction structuring and level 3: instructional content specification). Note that this hierarchical content structure does not reflect a specific instructional logic. This is specified with the episode editor. The author is left entirely free in specifying the decision model, wether learner controlled or program controlled.

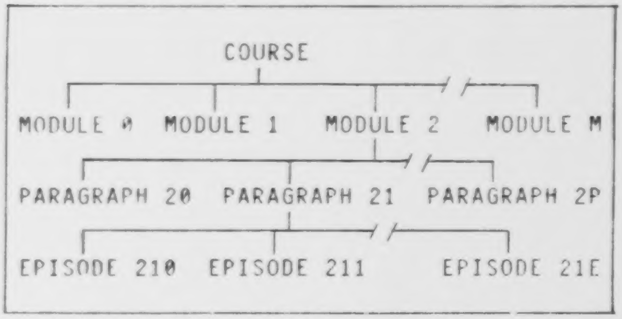
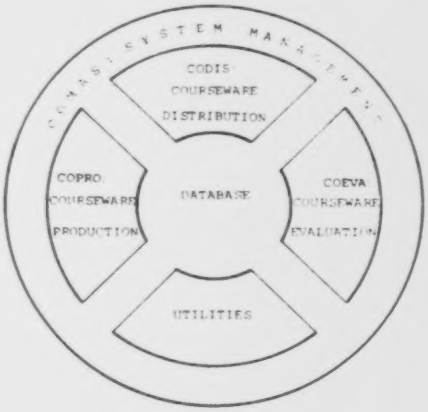


Figure 1: Course structure

STRUCTURE OF TAIGA

TAIGA consists of five sub-systems and some utilities configured as shown in figure 2:


COMAS: SYSTEM MANAGEMENT
CODIS COURSEWARE DISTRIBUTION
COPRO COURSEWARE PRODUCTION
DATABASE
COEVA COURSEWARE EVALUATION
UTILITIES

- COMAS	The COMAS subsystem manages acces to the other subsystems, performs elementary database functions etc. COMAS is not implemented as a seperate program, but is distributed over various other programs comprising TAIGA
- COPRO	COPRO is the subprogram used to create and edit courseware. COPRO produces two files per module, a 'program' file containing the instructional interaction, and a 'data' file containing the instructional content.

BEST COPY AVAILABLE

- CODIS	This is the subprogram students use to access the courseware. CODIS optionally stores student data (entry points, date/time tags, answers etc.) in log files for use by COEVA. User written procedures (PASCAL, FORTRAM, assembler) can be linked with the supplied CODIS object modules. These external procedures are called under courseware control.
- COEVA	COEVA accesses the student log files and produces overviews for evaluation purposes.
- UTILITIES	Several utilities are available, such as CODOC (Courseware Documentation), a program to produce a listing of the courseware.

THE COURSEWARE PRODUCTION SYSTEM COPRO

COPRO is the actual production system used for entering and editing courseware. The three layers for entering module content, instructional interaction and instructional content are called resp. the Module Editor, the Episode Editor and the Text/Graphics Editors.

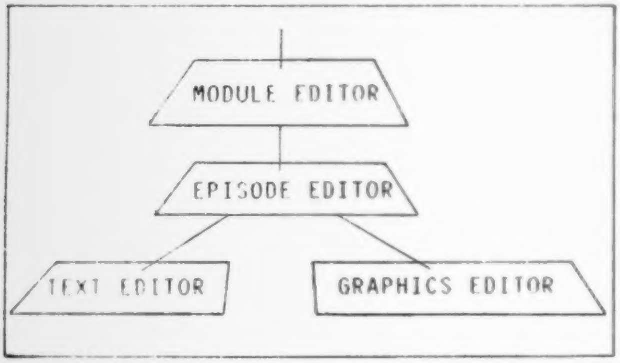


Figure 3 : Three layer structure of COPRO

THE MODULE EDITOR

After gaining access to COPRO from the system manager (COMAS), the author is prompted to enter the name of a module that is to be created or edited. If a module with the same name already exists this module will be linked for editing. If a new name is specified a new module entry will be created. This module initially contains one paragraph with one empty episode. See figure 4.

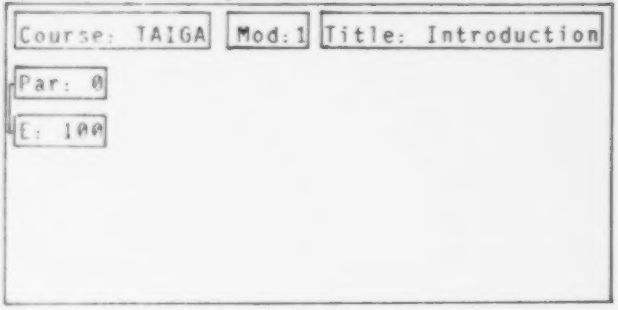


Figure 4: First level: Module/Paragraph Editor

The EXIT-key returns the author to COMAS. Pressing the EDIT-key when the cursor marks a paragraph entry allows entering a paragraph title. Pressing the EDIT key when the cursor marks an episode entry accesses the second level of COPRO, the Epsiode Editor.

LEVEL 2: THE EPISODE EDITOR

With the Episode Editor the instructional interaction can be defined using graphic building blocks called frames. At this level the screen looks like figure 5.

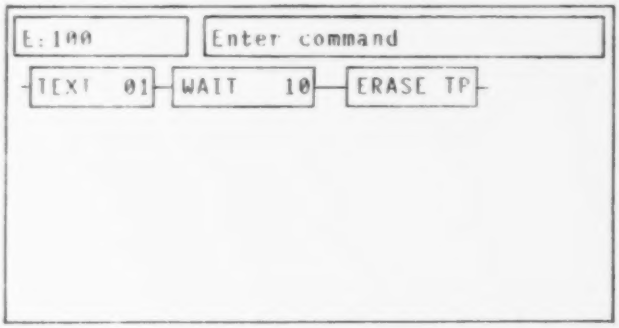


Figure 5: The Episode Editor

The screen contains 49 entry-points for entering frames. These points are organized as a 7x7 matrix. The cursor control keys are used for specifying the entry point. The INSERT-key followed by a single character key identifying the frame type causes the insertion of a frame. If parameters are required for which a default value can not (meaningfull) be specified, the system explicitly asks for them (e.g. branch parameters). The EXIT-key returns the author to the

Module/Paragraph Editor. Pressing the EDIT-key performs an action depending on the frame type. If the current frame contains instructional content (text, feedback, graphics) the system accesses the third level for the specification of this content. If the current frame has parameters for modifying it's basic operation pressing the EDIT-key allows editing these parameters (e.g. speciying conditions for test-and-branch frames). The delete-key will delete the current frame and shift back following frames.

SYSTEM VARIABLES

For purpose of answer processing and flow control a set of basic system variables is available: switches, counters, variables, textbuffers and keyword lists. Since most structured programs use only a limited amount of variables at global level, 26 variables of each type are predeclared. More exotic types can be declared using external PASCAL (FORTRAN) subroutines. The first letter of the variable defines the type and the second letter identifies one of the twentysix variables of that type (e.g. CX identifies Counter X). Variables are typed by use rather then by internal representation (boolean, integer, real, string).

ELEMENTARY BUILDING BLOCKS: FRAMES

Frames are the basic elements for constructing a course. At this moment 19 frame types have been implemented. The 20th frame (the VIDEO-frame for integrating interactive video) is 'under construction'. Frames serve one or more of the following functions:

- information presentation
- information input
- information manipulation
- information testing
- flow control
- miscellaneous

To give an impression of the frame an alphabetical list including a short description is presented here:

ASSGN pp	The ASSIGN-frame assigns a value to the specified system variable. Simple type expressions can be entered.
CHECK pp	The CHECK-frame compares the value of a variable with the result of an expression. Depending on the specified condition program flow continues with the frame connected to the TRUE or the FALSE output.
DOCUM	This frame is used to document the courseware. It can be edited like the text frame. CODIS ignores this frame.
ERASE pp	One of the predefined viewports can be erased with the ERASE-frame. In parameter mode the user can erase an arbitrary part of the screen.
FEEDB	The FEEDBACK-frame displays text in the feedback port on the screen. Text can be entered at level 3 with the Text Editor.
GRAPH	This frame renders access to the level 3 Graphics Editor to define (static) pictures.
HELP	Functions like an interrupt vector. When the student presses the HELP-key and HELP is enabled, flow control is passed to the HELP frame. Text can be entered with the Text Editor at level 3.
INPUT pp	Allows assignment of values to the system variables by the student. Edit-mode parameters control display colour, echoing, prompt character, position of input etc.
JUMP pp	Passes control to the specified frame within the current episode.
LOGFL pp	Writes specified data (system variable) to the student logfile, together with a date/time tag and an episode ID.
MULTC	Display multiple choice question (text entered at level 3) and waits for the student to press one of the four answer keys.
NEXT pp	Passes control to the first frame of the specified episode within the current module or passes control to the specified module. The value of the system

	variables is retained when a new module is loaded and executed.
OUTPT pp	Outputs the value of the specified variable to the screen. Display colour, position, font etc. can be specified in EDIT-mode.
PROGR pp	The PROGRAM-frame activates the specified external program. External programs can be written in MS-PASCAL, MS-FORTRAN or Macro-86. The TAIGA system variables are available to external programs.
QUEST pp	The QUESTION-frame compares a student answer with the specified keyword list. Several options control this comparation to allow flexible answer processing.
RETURN	The RETURN-frame is used in conjunction with the STORE-frame. Together they implement a subroutine mechanism. RETURN passes control to the specified episode whose identifier was stored by the last executed STORE frame. Subepisodes can be nested 8 levels deep.
STORE	Pushes the identifier of the current episode on the stack. This address is popped of the stack by the RETURN-frame.
TEXT	This frame contains the instructional content in the form of text. Text can be entered by positioning on this frame and pressing the EDIT-key which accesses the level 3 text editor.
VIDEO	Integration of interactive audiovisual media is considered an important 'feature' of an authoring system. This frame allows control of connected video equipment (VLP, VCR) in seperate screen mode or in overlayed screen mode. This frame is currently being developed and no final specifications can be given at the time of writing this paper.
WAIT pp	The WAIT-frame is used for timing purposes. It delayes execution for the specified period of time (in seconds) before passing control to the next frame. If the specified time is zero, the distribution-system displays the message 'Press CONTINUE-key...', waits for the student to press this key and then passes control to the next frame.

LEVEL 3: INSTRUCTIONAL CONTENT EDITORS

Level 3 consists of two editors, a Text Editor and a Graphic Editor. The appropriate editor is accessed by positioning the cursor at a text or graphics presenting frame (TEXT, FEEDB, MULTC, HELP, DOCUM, GRAPH) and pressing the EDIT-key.

THE TEXT EDITOR

When the Text Editor is entered the first time it asks the user if he wants default port size or not. If the user does not want to define his own portsize the default portsize is automatically set according to the type of frame from which the editor was entered. Using cursor control keys caracters can be entered at any position inside the port. Functionkeys allow insertion and deletion of characters and lines, storing and retrieving of text and control over the writing attributes.

Text Editor functions are:

- Insert character
- Delete character
- Insert line
- Delete line
- Exit to level 2 (Episode Editor)
- Store current line in buffer
- Load current line from buffer
- Toggle row/position counter (on/off)
- Zero attributes to default values
- Take attributes from current char.

Writing attributes:

- Underscore (on/off)
- Inverse (on/off)
- Set (number of soft character set)

THE GRAPHICS EDITOR

With the Graphics Editor simple graphic objects (lines, arcs, texts) can be combined into complex pictures. Objects can be moved and copied across the screen using a crosshair and the cursor control keys. Pictures can be stored in or retrieved from graphic libraries.

A short summary of commands:

- Insert object (Line, Arc, Pie, Text, Filled Area)
- Delete object
- Exit to level 2 (Episode Editor)
- Move object
- Copy object
- Label object
- Update object
- Store picture
- Load picture
- Edit drawing attributes

Some writing attributes for objects are:

- Underline (on/off)
- Inverse (on/off)
- Fillstyle (outline, solid, halftone, pattern)
- Line type
- Write mode (replace, complement, overlay and erase)

SIMULATION/GAMING: THE PROGRAM FRAME

As mentioned before, use of TAIGA does not require knowledge of a programming language or computer hardware (e.g. graphic protocols for terminals). For most forms of CAI (tutorial) the implemented set of frames has proved to be adequate. But implementing (quality) simulation and gaming courseware however requires detailed knowledge of computer hardware and software techniques and the ability to apply them. Implementing fast animation often requires programming in machine language. The demand of offering a high level 'user friendly' interface and the demand of allowing the implementation of complex software techniques as well as low level access to the hardware are contradictory. Two approaches are often found in authoring tools. The first approach is extending an existing (high level) programming language with CAI oriented procedures. The second approach is designing a dedicated authoring system with a high level programming language like instruction set. Neither of these approaches are satisfactory to our opinion. The first approach requires detailed programming knowledge even if simple tutorial CAI is made, the second aproach often lacks access to advanced system features. TAIGA follows a third approach by:

1. offering a high level interface to the instructional scientist who can define the instructional logic using the level 1 and level 2 editors,
2. offering a high level interface to the graphic designer who can use the interactive full screen text and graphic editors of level 3 for entering instructional content, and
3. offering a low level interface to computer scientists, allowing implementation of additional procedures in a well documented, and widespread language like PASCAL, or even machine language.

CONCLUSIONS

The Educational Centre has about three years of experience with TAIGA in a wide range of CAI applications, both internally (university programs) and externally (cooperation projects with other universities and high schools). Most of the bugs have been worked out now and we have quite satisfied users. Future developments will include integration of overlayed interactive video and Macintosh like menu/mouse control.

THE PLAN
An Instructional Systems Design Approach to the Analysis and Design of Technical Documentation

Patricia Lawson
Microtel Learning Services
1795 Willingdon Avenue
Burnaby, B. C. V5C 5J2

ABSTRACT:

A goal of technical documentation is to explain and reinforce, if not educate. Technical writing models, therefore, should incorporate elements of instructional design, especially in the planning stages. This article examines this writer's application of the analysis and design phases of an instructional design model to her writing model, and gives practical, concrete examples for doing so.

In many technical writing projects, a technical writer's goal is to explain and reinforce, if not educate. My written product can be only as effective as my definition of its instructional purposes and the learning characteristics of my readers. For this reason, I use a modified instructional systems design (ISD) model to produce my written products. This improves their instructional quality and their usefulness to the end user.

The ISD model has five phases: analysis, design, development, production and evaluation. I describe my use of the ISD analysis and design phases in the technical writing process.

ANALYSIS

An ISD analysis has four basic outcomes: 1. the preliminary study, 2. the job analysis, 3. the population analysis and 4. the analysis of training needs and job aids. These determine the necessity of the proposed training, identify the content and audience, and produce an initial estimate of the time and money. I use them to define: 1. the purpose of the documentation package, 2. the primary and secondary readers, 3. the primary reader's characteristics, 4. the existing documentation, and 5. the scope of the writing project.

1. Documentation Package Purpose

This information comes from the ISD preliminary study and the job analysis. It is a broad statement of the kinds of material, e.g. reference, procedural or tutorial, to be covered. It indicates the number and types of manuals in the total documentation package and helps to focus design decisions regarding tone and style.

2. Primary and Secondary Readers

This information comes from the population analysis. The primary readers are the end-users. However, other groups of people will access the documentation package and judge your company based on it. These are the secondary readers. They include marketing staff and the end-user's supervisor(s), co-workers and family. Design decisions should consider this larger group. For example, you might decide that although the manual is for conservative accountants working in separate cubicles, your company name on the cover should be readable from across the room so that everyone who enters that cubicle will notice it on the accountant's shelf.

3. Primary Readers Characteristics

I define the primary reader using the five learner characteristics defined by the ISD population analysis: entry level requirements, environmental familiarity, language capabilities, reference frame and motivation. This definition is the most useful part of the ISD for me. As much as possible, these must be identified and considered because they directly affect level, pace, organization, approach, and tone and style, both of the text and the illustrations.

a) Entry Level Requirements

Entry level requirements are what the readership can be expected to know before using the materials. Based on the ISD model, I identify these in terms of acquirement and accomplishment.

Acquirement is what a person already knows and accomplishment is what he can learn to do. For example, "One may wish to find out whether a person knows how to measure voltage with a certain type of oscilloscope. Even if he fails to do (accomplish) this ..., he may still have learned [acquired] most of the principles and skills involved .... It may even be that he is just unfamiliar with the type of oscilloscope used ...." (1)

In determining how to reach my reader, I decide which is needed: a series of steps and procedures to accomplish the task and/or an explanation of the environment and tools to acquire the necessary peripheral information. For example, should an operating manual for a computer include explanatory information on the importance of making back up copies of disc or diskettes as well as procedures for doing so?

b) Environmental Familiarity

On the other hand, I cannot assume that my readers are familiar with the physical environment where the manual is to be used. For example, are they comfortable in a computer-based training classroom; do they know how to use the system? If not, descriptive paragraphs and/or illustrations may be required before the "actual" content is presented. You have to weigh "how much is enough" versus "how much is cost-effective".

c) Language Capabilities

It is important to assess the reader's ease with the english language as well as their reading level. If most speak english as a second language, then my word usage should be "dictionary" correct and the style formal. Conversational language increases the number of words and colloquial terms obscure meaning.

d) Reference Frame

It is also important to assess the reader's understanding of both technical and non-technical terms and concepts. Accepted "lingo" in one part of the country is not necessarily used in another. Also some terms are used so much, they lose their meaning, e.g. the ubiquitous "system"

Effective manuals also work from symbols familiar to their readers. You must be careful not to assume visual familiarity. I try to identify and work from this known frame of reference. I try to use the same written and visual images and symbols my reader uses.

e) Motivation

These first four characteristics affect how readers learn. Motivation relates to how readers feel about content or having to read it. i.e., what's in it for them? Contributing factors include reader age and company reinforcement of reading on the job. For example, an older, general-interest group needs to relate new ideas to older, established symbols and images they are familiar with. Similarly, readers whose company does not give them time on-the-job to read need manuals that are very portable, accessible and to the point.

4. Existing Documentation

I find out how much and what kind of documentation already exists. This helps me define the amount of research to be done and the learning curve for the writing team to become comfortable with the product. I always want to know what already exists so that I can "canabalize" it.

5. Scope of the Writing Project

I think about what could be written and ask myself "how much do I want to cover?". I set a limit on the amount and kinds of research to be done. I then ask myself "what points do I really want to make; very roughly, what do my readers need to know?" I try to estimate the number of pages of finished text the project will have.

This is my version of the ISD analysis of training needs and job aids. From it, I broadly estimate what needs to be done, how long it will take and how much it will cost. This estimation includes production requirements. It is a preliminary document only, established as a basis for negotiation and starting point for the design tasks.

DESIGN

An ISD design has four basic outcomes: 1. the determination of objectives, 2. the design of tests, 3. the verification of tests and objectives, and 4. the preparation of a design document. Of these, I ldapt the form and content of the first and last to produce my design document(s).

There are four design steps in my writing model. These are to write:1. a statement of purpose, 2. learning objectives, 3. a design document and 4. a table of contents. Each of these tasks must be done for the overall documentation project, each manual and each section within each manual, a total of twelve tasks. These tasks are inter-related and often concurrent.

Based on my reader definition, I list the manuals required. For example, as the producer of documents to support computer-based training courses, I might produce a course-specific instructor's guide, student reference notes and marketing brochures to accompany a generic instructor's manual and student's handbook on the computer system.

1. A Statement of Purpose

I write a statement of purpose for each manual and each section. These relate to the statement of purpose for the entire project, done in the analysis. These statements should answer: why the manual is needed, who uses it, what it covers/does not cover, how it relates to the entire package and what the sources for concrete examples are. The result states the relevance of the material. It helps to focus the writing effort and eliminate nice-to-know but superfluous information.

2. Statement of Learning Objectives

I then write a brief one or two paragraph statement of learning objectives for each level of the documentation - what is the readership expected to know or be able to do after reading this section, manual and package? The statements of purposes and objectives provide criteria for reviewers to judge the drafts - does the piece cover and do what it says it will? I then write a design document for each manual.

3. The Design Documents

I rank the manuals based on priority, complexity and preparation time, and estimate their length in pages. This refines the estimate done in the analysis.

Design decisions should be documented to provide a written track record and as a basis for negotiation and review. Depending on the scope of the documentation project, I write one for the project as a whole and one for each of the manuals. Table One shows a generic design document that I use for both the overall project and for each manual. I change the form and content as necessary and write an update for revisions. I often include a sample page layout for each of the manuals.

The design document specifies tone and style. Part of style and tone decisions is the use of technical terms and 'jargon'. Technical writers usually avoid both because they can be non-precise and limited in their application. However, they often say in one-word for a specialized audience what would take several for a more general group. If the reader is specialized, it makes sense to use their "language"

4. Tables of Contents

Tables of contents fall out of the statements of purpose and objectives. First I write an abbreviated table of contents for each manual, then a detailed table of contents for each manual and each section. The detailed tables of contents list points under each sub-heading and suggestions for illustrations. If sufficient detail exists (and if the reviewers don't make significant changes), then the writing process can be as simple as filling in the subjects and connectors.

SUMMARY

All writing models undergo similar processes to a some extent. Table Two compares an ISD analysis and design model to the same phases of my writing model. Because I believe that all technical writing is, at some level, educational, I suggest that technical writers explore instructional system design models and incorporate those elements of them which make their written products more instructionally sound.

Table One

PROJECT NAME
SPECIFIC MANUAL NAME
DESIGN DOCUMENT
Date

Goal: The goal may be instructional and/or as a marketing thrust. The goal is from the producing company's point-of-view and should be transparent to the end-user.

Nature of Technical Writing Task:
Is this essentially an editing (what kind - structural, content?) and/or formatting job to be done as quickly as possible or is it a full-scale writing, illustrating and/or coordinating job?.

Content:
This should be a one or two paragraph statement of what will be covered.

Time Line:
List the tasks or refer to another document.

Manpower:
List the personnel involved, including Project Lead, Writers (Lead Writer, Support Writer(s), Content Editor, Copy Editor), Illustrators, Clerical Support, Typesetting and Printing resources, etc.

Readership: Primary
Secondary

Purposes: Primary
Secondary

Instructional Objectives:
- What should the readership know or be able to do after reading the manual(s)?

List of Documents:
- and/or their component parts

Tone/Style:
- What is the relationship between the writer(s) and the reader?
- What attitude does the writer(s) take towards the subject matter?
- What is the reading level?

Design:
- page size
- covers
- binding
- typeface, typesize & typespacing
- print single sided, double sided
- ?lb. paper
- ? color type on ? color paper
- glossary ?

Table One cont'd

Format:	- heading - words, position, typesize & face - sub-headings - as above - footing - as above - rule under heading & above footing - justification - indentation
Layout:	- maximum line length - margins

Table Two

TDG	MY APPLICATION
ANALYSIS	
1. Preliminary Study - effectiveness of training as solution	1. Preliminary Study - purpose of overall documentation package & its components
2. Job Analysis - identifies skills & knowledge to do a job	2. Job Analysis - basis for steps & procedures - identifies existing written sources
3. Population Analysis - identifies target population in terms of: - content assumptions - language capabilities - reference frame - environmental familiarity - values & learning style	3. Population Analysis - the same
4. Analysis of Training Needs & Job Aids - time & money estimates	4. Analysis of Manual Requirements - what needs to be done; how long it will take & much it will cost
DESIGN	
	1. Determining Purpose - overall documentation project - manual-specific - section-specific
1. Determining Objectives - performance statements	2. Determining Objectives - overall documentation project - manual-specific - section-specific
2. Design Document	3. Design Document - overall documentation project - manual-specific - layout sample(s)
	4. Table of Contents - list of manuals - manual-specific - section-specific

REFERENCES

1. Training Development Guidelines, (Geneva, Switzerland: Interregional UNDP/ITU project CODEVTEL, 1979), p. 341.

A MODEL FOR SUPPORTING UNIVERSITY CBI PRODUCTS AND SERVICES

Paul L. Hyde
Deborah G. Mellor
Cynthia A. Parker
Catherine B. Phillips
James H. Wilson

Office of Computer-Based Instruction
University of Delaware

ABSTRACT

One successful model for interaction between CBI development projects and people outside their institution involves a team specifically charged with the responsibility. Members provide and coordinate services of the University of Delaware's Office of Computer-Based Instruction. The tasks included in these services are outlined, and estimates of the time necessary to accomplish them are provided.

INTRODUCTION

Comprising the services offered by the University of Delaware are a variety of functions associated with the development and delivery of CBI courseware. This paper will 1) summarize the background of one services organization, 2) describe the services it offers to both PLATO® subscription customers and those attending demonstrations and workshops, and 3) tell how it handles sales of hardware and courseware products.

BACKGROUND

Several universities have met the challenge of distributing software that has been developed within their institutions. Software developed at the University of Waterloo is distributed through Watcom Products, a company owned in part by the University. The company employs a staff of eight and distributes about forty products. Stanford University distributes software through a more limited staff and makes a wide variety of specialized materials available. Some parallels exist between these efforts and those of the University of Delaware to distribute CBE courseware materials.

Increasingly specialized structures are often a by-product of growth. When serious CBI work began at Delaware in the mid-1970s, a small staff dealt with all of the necessary functions--assisting faculty in reviewing materials, supervising student use, developing new materials, repairing terminals, and consulting with others both more and less experienced. This model worked well, but showed strain as the number and diversity of projects and successes grew. It has the advantage of establishing a staff knowledgeable in all areas and able to deal with all or most aspects of a situation without delay for communication. Its disadvantages include 1) difficulty finding fully qualified staff and training them for all tasks, 2) high overhead for staff members required to stay abreast of all areas and participate in their development, 3) the possibility of lacking specialized expertise to deal with a need, and 4) the lack of a clearly identified contact for a purpose.

An important catalyst in the development of a services unit was the University's acquisition of a PLATO mainframe. Because the smallest machine available had a capacity greater than the University's needs at the time, the excess capacity was sold to outside users to help pay for the cost of having and running the machine. After the first few outside users joined the system, the University's Provost ruled that for every substantial increase in stable outside income the Office could add an additional staff member. Figure 1 shows the growth of the OCBI services staff in relation to the number of PLATO subscriptions sold. The Office, in turn, made certain to allocate these staff members as necessary to assure adequate support to the effort to serve new and existing customers. Figure 2 shows the current staff allocated

PLATO® SUBSCRIPTIONS V. SERVICES PERSONNEL ALLOCATION

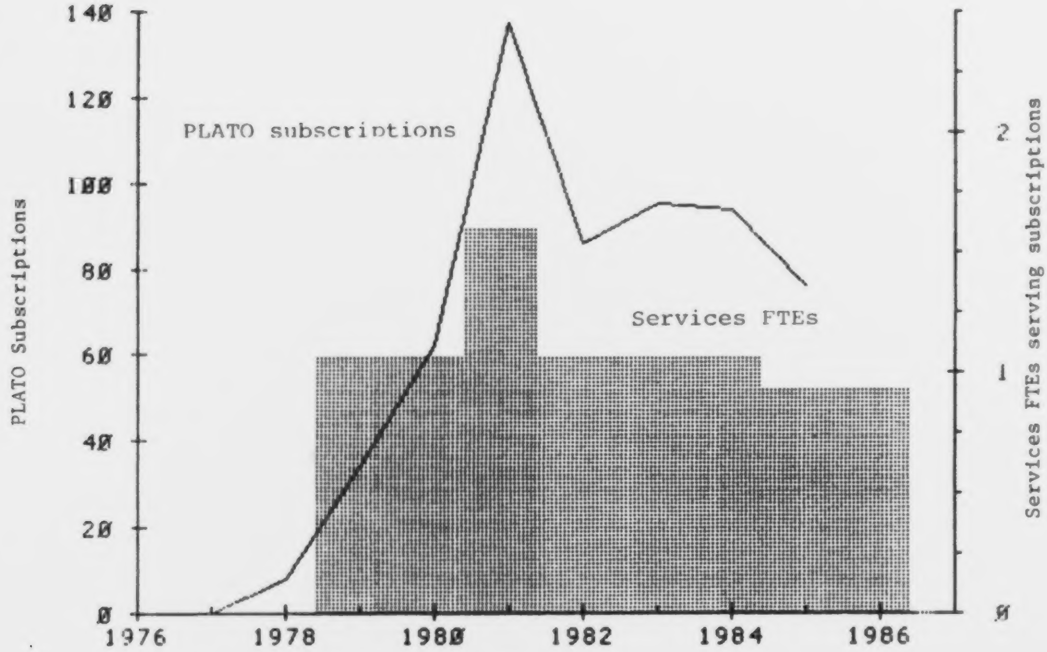


Figure 1.

CUSTOMER SERVICES PROVIDED BY OCBI

Services Personnel - hours per week (% of time allocated to performing services related activities)	PLATO® Service Subscriptions	Microcomputer Software and OCBI Products	University of Delaware PLATO Lessonware	Computer Demonstrations	CBI Workshops	Equipment Loans	Visitors
Associate Director for Operations (30%)	4	4	0-1	0-1	0-1	0-1	0-1
Senior Customer Services Specialist (40%)		8	4				2.5
Senior Customer Services Specialist (100%)	1	25					2.5
Programmer/ Analyst (70%)	25						2.5
Jr. Programmer/ Analyst (25%)		2		2	0-1	1	2.5
Total Hours	30	39	4-5	2-3	0-1	1-2	10-11

Figure 2.

PLATO® is a registered trademark of the Control Data Corporation

BEST COPY AVAILABLE

to the services unit and the total number of hours contributed by each.

Concurrently, other organizational changes took place, establishing a unit to consult with both inside and outside users on questions related to programming and to the PLATO system, a unit to deal with equipment problems of any sort, and a unit to deal with system operation, programming, and datacommunications. Although not performing these functions directly, the services unit works closely with those who do to meet the needs of outside users. Responsibilities such as cataloging and arranging visitors' agendas seemed natural for the services unit initially, and others were added as needs grew.

SERVICES

SUBSCRIPTIONS

As previously stated, excess capacity on the University's PLATO mainframe is sold to outside users to partially defray the expense of having and running the machine. The OCBI services staff helps potential customers determine whether their CBI needs can be met via the PLATO system and, if so, how many subscriptions are necessary to meet those needs. Once need is established, a contract is signed by the customer and the University for subscription service. Services personnel offer advice at this time regarding the use of dedicated vs. dial-up telephone lines and equipment necessary to access the system, although the customer bears the responsibility of acquiring equipment and datacommunications lines. Once hardware is in place, an account is set up on the system by a services representative, and arrangements are made for an on-site training session. During this session, a services representative will "walk" a customer through accessing the system, lessonware, course records, and notes. The representative is also responsible for initiating the invoicing procedure for subscription service according to the terms of the customer's contract.

The University of Delaware's subscription service includes access to courseware available in the University's lesson library. If they desire, customers may contract for additional courseware that does not reside on the system. A customer is entitled to active memory space, file parts of disk space, and archive rights. The University also rents additional file space to those desiring it. Customers enjoy unlimited use of available processing capability. OCBI has staff readily available for communications, courseware, and operational problems. Should customers experience any problems, they contact their services representative who acts as liaison between the customers and other OCBI staff (such as programming consultants) in the resolution of the problem.

Offering subscription service requires personnel for communicating with customers and keeping records of file space, invoicing, contracts, and sales. Time devoted to accomplishing services tasks varies greatly from week to week and month to month due to the variety found among outside customers with regard to system use, fiscal years with different starting dates, and contract types. An average of thirty hours per week of services personnel is necessary to offer subscriber service to OCBI's current fourteen customers (representing eighty subscriptions). Additional Office support time is necessary from programming consultants and from administrative, systems, maintenance, and secretarial staffs. These OCBI personnel resources are required to achieve and maintain an acceptably high level of service.

DEMONSTRATIONS

The services staff at OCBI is responsible for demonstrating OCBI courseware at national and local conventions, computer fairs, and software expositions. As a first step, courseware must be selected based on the expected composition of the audience and the availability of staff and space for the exhibit. Next the initial reservation and space request must be made and followed through with the furniture, electrical, and telephone requirements. Volunteers are solicited from the entire OCBI staff, and a schedule is made for people to demonstrate courseware at the booth during exhibit hours. Those staff members who are not familiar with the equipment or courseware being exhibited are given instruction about what is appropriate to say. Finally, the moving and configuring of the equipment is coordinated. The time spent preparing for these events ranges from two hours to eighty hours of services staff time, plus usually an equal amount of time from support personnel. The amount of time depends on the size of the exhibit, the length of the convention, and its distance from campus.

EQUIPMENT LOANS

The services staff accommodates faculty and staff members who desire off-campus equipment loans to make presentations of OCBI courseware. A form is available for interested people to make their equipment requests. Once this is submitted and approved, the services staff makes sure all of the arrangements are carried out. Since the maintenance group is responsible for equipment locations and inventory, it works with services on moving equipment off-campus.

WORKSHOPS

The services staff also organizes training workshops. These workshops, which are available on site or on campus, are customized to meet the needs of a client or group of clients. Topics have included the following: introduction to the features of CBI systems, how to design and program PLATO and microcomputer lesson materials, how to integrate CBI into an existing learning environment, and how to manage CBI resources. The services staff coordinates the logistics of the workshop, while the teaching is handled by an OCBI staff member who is an expert in the subject area. A brochure has been produced by the services staff to acquaint potential customers with workshop and consulting services.

VISITORS

Over the past ten years, OCBI has hosted an average of sixty-five visiting colleagues each year. It takes ten to fifteen hours for a services staff member to arrange, prepare for, and host a one-day visit, which can include tours of computing sites, time using terminals, and meetings with University personnel. Hosting responsibilities are generally alternated among the staff so that no one's time becomes overcommitted to visits.

COURSEWARE PRODUCTS

GUIDO® EAR TRAINING SYSTEM

The GUIDO Ear-Training System is one of Delaware's first development projects and marketing efforts. The package of five lessons is available through OCBI in mainframe or Micro PLATO versions. Customers obtain the lessons through license agreements drawn up by the University. Mainframe customers receive a tape of the package and are billed for hourly usage, which is reported by the customer. Micro GUIDO customers contract for either a six-month or two-year unlimited usage license. They receive one combination instructor/student floppy disk per learning station. OCBI will make additional copies of the disks upon request. Updates of software versions are mailed to customers when the lessons are revised. It is estimated to take five hours per week to accomplish the tasks involved in this service: contract and paperwork, handling inquiries, customer questions and concerns, and carrying out marketing ideas.

Designed to accompany the five GUIDO lessons, the University of Delaware Sound Synthesizer was developed and built by OCBI. Introduced at the 1981 ADCIS conference, the UDSS is sold for a one-time fee. It comes with a ninety-day warranty, after which repairs can be made by the OCBI hardware development group on a time-and-materials basis. Updates of the ROM chips, with installation instructions, are sent to the customer free-of-charge. Typically it takes three hours per week to handle the paperwork involved, to address customer questions and problems, and to implement our marketing effort.

CENTRAL SYSTEM LESSONWARE

In 1982, the University of Delaware began efforts to publish and distribute its lessons independently, while continuing to submit lessons to Control Data Corporation's Authors' Library. As of July 1, 1985, forty-seven mainframe lessons have been published by OCBI and were available to other PLATO systems. After a lesson has passed OCBI's internal publication review and is listed in the catalog, institutions wishing to use it send a purchase order to OCBI. A Central System Lessonware License Agreement is drawn up. A tape and accompanying documentation is then sent to the customer, who is responsible for keeping and reporting to OCBI the number of hours of use in each quarter. The customer is billed a nominal charge per contact hour. Should the customer report that the lesson does not work as documented, Delaware will make the necessary corrections and send a new tape. This service is estimated to take four hours per week to handle paperwork, catalog updates, and customer questions.

VIDEODISC MUSIC SERIES

Supported by a grant from the National Endowment for the Humanities, the University of Delaware has produced the Videodisc Music Series, featuring performances from fourteen institutions. The package includes four two-sided discs and a printed anthology, and it sells for a one-time fee.

The series includes full-color video recordings of ten musical masterworks from the classical and romantic periods. The discs were pressed by Pioneer Video and are distributed by OCBI.

Over the past two years, the role of the services staff has been to act as the customer contact, to answer any questions, and to keep customers informed of the proposed distribution schedule. These functions take one to two hours per week. The services staff will handle distribution of the discs when they become available. It is estimated that distribution responsibilities, locating packaging materials, filling and mailing orders, will require an additional five hours per week.

MICROCOMPUTER PRODUCTS

LATIN SKILLS

In August of 1984, OCBI began to distribute Latin Skills: Five Programs for Microcomputers, its first microcomputer software package developed for delivery on a machine with a broad base in the educational market. Twelve months prior to the distribution of this package, the services staff began making preparations for the product's release. The first several months involved planning and decision making: designing promotional materials (a brochure, order forms, advertising materials, and various pieces of artwork for the packaging), procuring the packaging materials, and arranging for commercial disk-copying services. These activities required an average of ten hours per month. Three months prior to distribution, an intensive effort was required to prepare the package documentation and send it to the printer, to coordinate the bulk mailing of brochures, to respond to the increasing number of inquiries from potential customers, and to prepare the product package for distribution. One services person devoted thirty-five hours per week for most of the three months to accomplish these tasks. Additionally, support staff contributed ten hours per week during that time period. Now that the product is released, one services person is occupied ten hours per week, and one clerk contributes ten hours per week to provide the customer support for this package, which on the average generates three purchase orders and five to ten letters of inquiry and telephone calls each week. Customer support for this package, as well as for most products distributed by OCBI, includes arranging advertising, setting up demonstrations, monitoring inventory, and maintaining up-to-date income and expense records for the product.

LIBRARY RESEARCH SKILLS

In January of 1986, Library Research Skills: Four Programs for the IBM® Personal Computer was released, and distribution of the package began. As with Latin Skills, ten months prior to distribution the services staff began the process of readying the package for marketing. Following the precedents established by Latin Skills, one person devoted eight hours per week to designing promotional materials, procuring packaging materials and coordinating the printing of the product documentation. Customer support for this package is provided by one services person contributing eight hours per week and by one clerk devoting three hours per week.

CONCLUSION

Although CBI units are highly diverse in their needs for the tasks outlined above, nearly all of them face needs for some combination of them. While meeting these needs with programmers, designers, or faculty members alone may have some advantages as a model, the use of a specialized team at the University of Delaware has been appreciated by those served by it and represents a viable competing model. The amounts of time spent and measured under this model may be small compared to time spent and not measured under other organizational schemes. Moreover, the opportunities lost if this time is not spent may be more costly than the time itself.

IBM® is a registered trademark of the International Business Machine Corporation

GUIDO® is a registered trademark and service mark of the University of Delaware

USER-ORIENTED TOOLS FOR CBI DEVELOPMENT

George A. Reed

Office of Computer-Based Instruction
University of Delaware

Abstract

The Office of Computer-Based Instruction at the University of Delaware has produced an integrated set of authoring tools for the IBM PC. These tools are written in Pascal and provide an easy, user friendly environment for the development of lessons. This environment provides many advantages over other common development methods. The development and use of such tool collections should be considered as a viable alternative to the use of complete authoring systems or educational authoring languages.

Introduction

The Office of Computer Based Instruction (OCBI) at the University of Delaware has been producing Computer Based Instruction (CBI) for eleven years, first on the PLATO system in the Tutor language, more recently on a VAX 11-780 using the Digital Authoring Language, and on a variety of microcomputers using several different languages. Both mainframe environments used at the OCBI have extensive collections of authoring tools to support CBI development. When development began on microcomputers the lack of this support was very evident. Our first efforts at CBI on micros proved much more difficult than authoring on the mainframes had been. Authoring on the micros has become progressively easier as new tools have been developed--in fact, the ease of CBI authoring on microcomputers can now approach that of mainframes.

The IBM PC Authoring Environment

After considering several languages, MS DOS Pascal was selected as the language for our first IBM PC lessons. The reasons for choosing Pascal to develop CBI are still valid and include:

- -Wide user base by both CBI and non-CBI developers.
- -Ready pool of trained Pascal programmers.
- -Pascal support by third party hardware/software vendors.

As the first lessons were developed, we found that we were also, by necessity, writing tools to aid in lesson production. It was soon apparent that an organized effort was needed to develop a good, integrated authoring environment. The first question that we needed to answer was this: Should the environment take the form of a set of independent tools or of a complete authoring system?

Even though the thought of a complete, non programming authoring system is very attractive, we decided that no authoring system, either commercially available or "home grown" could supply our needs. An authoring system is well suited to the novice user who has little development support or to a user doing fairly standard CBI. Authoring Systems not generally suited to an academic environment where the needs and specifications of each subject matter expert differ widely and where CBI developers tend to be innovative and to take advantage of the newest technology.

In an academic environment, software developers are always doing something new, trying new approaches and experimenting with new hardware. We decided to develop a set of authoring tools with the philosophy that they should aid in the production of CBI but never limit the innovation or imagination of the author.

A set of tools can more easily be adapted to changing hardware and software better than an authoring system or most dedicated CBI authoring languages. New software products, of interest to an educator, are more likely to interface to Pascal than to other languages, making a Pascal tool set more useful. For example, IBM's Topview Programmers Toolkit currently has only Pascal and Assembler libraries. Hardware vendors are also likely to support Pascal before many other high level languages. An integrated set of Pascal tools also provides the user with flexibility to use other languages such as Fortran or C for parts of their program--these languages can easily be interfaced with Pascal.

Another important advantage of tools is that the graphics produced can be based on device drivers. With a device driver, any programs using the graphics are independent of any given display adapter. When run on a display that provides increased resolution or color, the application program will automatically take best advantage of the available capabilities. When a new display device (for instance the IBM Enhanced Graphics Adapter) is marketed, only the device driver needs to be written so that existing tools may use it. So an application can be written for the Tecmar Graphics Master with 640 by 400 pixel resolution and still be run on the IBM Enhanced Graphics Adapter with 640 by 350 pixel resolution. This transportability is a tremendous advantage over traditional authoring languages.

The tools developed so far have allowed us to cut the time for development by more than one half without sacrificing the flexibility to do what we really want.

The Tools

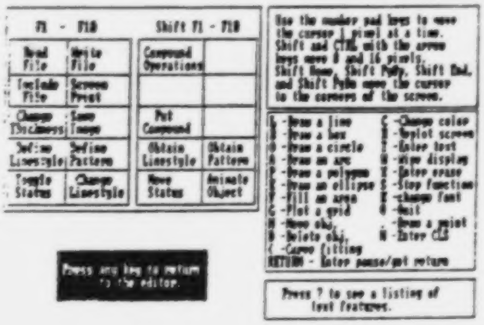
Graphics Editor--Much of the time and effort spent in developing CBI consists of positioning text and graphics on the screen. The simplest and most obvious tool required for the CBI developer is a graphics editor.

The OCBI Graphics Editor (GE) is an easy frame oriented editor that, unlike most graphics editors, produces Pascal code. This is in keeping with the philosophy that our tools should help and never limit what we can do. Code produced by GE can be modified or enhanced by anyone who knows Pascal. Thus the user is not limited to doing only what the editor allows.

The graphics produced by the editor are based on a commercially available graphics device driver. A display created using these routines can be shown on any display device that has a compatible driver written for it. In addition, the display will take advantage of the maximum resolution available on the current display device. Graphics displays now supported by the Graphics Editor include: IBM Color/Graphics Adapter, Tecmar Graphics Master, Hercules Monochrome Graphics Card, IBM Enhanced Graphics Adapter, AMDEK Color Graphics Card, QuadColor Graphics Card, and others.

Because the displays created with the editor can be shown without being part of a completed program, graphics frames can be created, reviewed, and revised many times without the time-consuming step of compiling.

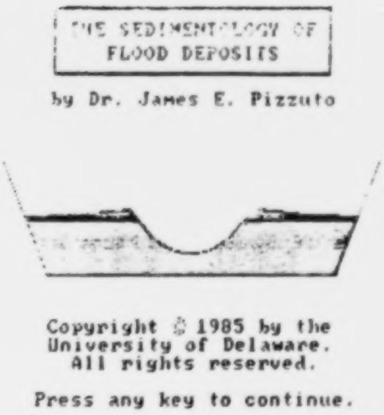
The GE supports the creation of text and graphics displays in both 80 and 40 column mode and the creation of text and character graphics on the IBM monochrome monitor or in text mode. A list of some of the options available in graphics mode is shown in figure 1. A lesson display created in graphics mode is shown in figure 2. A monochrome, text only display is shown in figure 3.



Use the number pad keys to move the cursor 1 pixel at a time. Shift and CTRL with the arrow keys move 8 and 16 pixels. Shift Home, Shift PgUp, Shift End, and Shift PgDn move the cursor to the corners of the screen.

FIGURE 1. Graphics Editor Options.

BEST COPY AVAILABLE



THE SEDIMENTOLOGY OF
FLOOD DEPOSITS

by Dr. James E. Pizzuto

Copyright © 1985 by the
University of Delaware.
All rights reserved.

Press any key to continue.

FIGURE 2. Color Graphics Display.

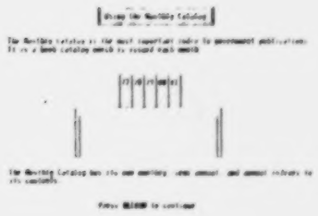


FIGURE 3. Monochrome Display.

Other capabilities of the GE include:

An interface to a Pioneer LDV 1000 through the Video Associates videodisc interface adapter. This interface allows the user to create text and graphics overlaid on a video image and to store the screen along with the image frame reference for later use.

Use of several mice and a digitizer tablet for graphics entry.

Character Set Editor--The teaching of languages and the production of animations, among many others applications, require the use of special characters and shapes. The character set editor makes it easy to produce, store, and call back multiple sets of special characters on the same display. These characters plot at the same rate as text characters and can be used in GE with the same special attributes as are applied to text--sizing, color, etc.

The character set editor allows multiple characters to be edited simultaneously so that pictures can be created that can later be called back to the screen through keypresses. The special characters can be assigned to keys typeable by students allowing them to type correct foreign language characters. As a starting point for the user, existing character sets, such as the standard IBM character set, can be loaded into the editor for modification.

Judging Routines--Pascal routines have been written to perform most of the common judging of student input. These routines are very general and can be copied by programmers and used as templates for their own students interaction. Some of the routines available include:

String judging, which compares the students input to the "correct" answer or answers. This routine is used for menus, multiple choice or true/false questions, and one or two word open input.

Math judging which accepts a numerical answer. This routine allows expressions using common mathematical operations. The expressions are evaluated and the resulting value is written to the screen prior to judging.

Screen position judging which reads the cursor position from the screen. This routine is useful for cursor key or mouse based interaction.

Each of the judging routines allow for several levels of feedback for incorrect answers before the correct answer is given. They also provide for multiple correct answers and anticipated wrong answers with special feedback. These options cover most of the usual CBI needs and, because of the open ended nature of the tools, more options can be developed as the need arises.

Future Plans

Recently, the Graphics Editor has been expanded to support Interactive Videodisc (IVD) capability using the Video Associates Corp. Videodisc Interface. Work is now underway to integrate a

BEST COPY AVAILABLE

commercial database program to the editor so that authors will be able to access a videodisc's contents according to parameters such as frame title or subject matter. This indexing will allow the author to sort a set of video frames, from a disk containing a much larger number of frames, and reference them as though they were the only material on the videodisc.

Other plans include the addition of more generalized open ended answer judging capabilites than are currently possible using Pascal. Intelligent answer judging, written in C, could be interfaced with the Pascal routines and be called by non-C programmers.

Conclusion

Tools such as those developed at the OCBI are useful even for experienced programmers doing complex lessons because they speed tedious tasks and allow for a uniformity of style. They also ease the job of less skillful programmers and allow them to do more complex and smoother lessons than they could otherwise do.

Although much is being said about the usefulness or necessity of complete authoring systems--today with the changing needs and abilities of instructional strategies and hardware systems a good, general purpose language with an integrated collection of tools is much more useful than a static AS or CBI AL.

NEEDS ANALYSIS FOR A COMPUTER-BASED STATISTICS COURSE

Mary Jac M. Reed

Office of Computer-Based Instruction
University of Delaware

ABSTRACT

The University of Delaware's Office of Computer Based Instruction, under a grant from the Digital Equipment Corporation, is developing a one-semester, computer-based statistics course. This paper will specifically detail the planning and design for the statistics package as an example of a needs analysis performed for a computer-based course.

When the Office of Computer-Based Instruction (OCBI) at the University of Delaware began the development of a computer-based statistics course, under a grant from Digital Equipment Corporation, the goal was to create a flexible, well-planned, and user-oriented statistics tutorial package. The modules are programmed in the Digital Authoring Language (DAL), a computer-based educational (CBE) language, on a VAX minicomputer running under the VMS operating system. The lessons are delivered via a router and record-keeping system, the Courseware Authoring System (C.A.S.), that allows easy student access to the modules and easy instructor access to student data. This paper will discuss some of the features that were incorporated into the computer-based course to achieve the level of flexibility and usefulness desired.

Before beginning large development projects at the University of Delaware, a needs analysis addressing the following points is performed.

1. Is there a need for CBE materials in this subject area?
2. Will the CBE component be stand-alone instruction or supplemental ?
3. What is the target population and how many potential users are there?
4. Does the subject matter lend itself to self-paced, individualized instruction?
5. Can the materials be adapted to many different populations within the target audience?
6. Is the content stable or does it become outdated quickly?
7. Does the subject area suggest graphical or visual displays, rather than text?
8. Can adequate record keeping data be kept on student performance?

THE NEED

When the analysis was conducted for statistics, it was found that this content area could benefit from CBE tutorial materials. There were many computerized software tools to help in the analysis of data but there were few instructional lessons that taught statistics concepts. The faculty at Delaware felt that tutorial statistics lessons would provide good individualized, self-paced instruction to beginning statistics students in many different courses. Additionally, CBE materials are an excellent media to provide graduate students, faculty, and career professionals with short courses on specific statistics topics that they may need to learn or review.

STAND-ALONE OR SUPPLEMENTAL

The decision next has to be made whether a complete stand-alone course or just supplemental exercises will be developed.

At the university, more than twenty introductory statistics course are taught with as many books by several different departments. Each department's course emphasized different areas of statistics.

No single textbook could be identified to which to 'key' the CBE modules. To adapt to this challenging situation, it was decided to develop a generic course comprised of common introductory topics that could be made applicable to a large number of disciplines, textbooks, strategies, and could be made independent of a formal course. Strategies employed to this end were the inclusion of a variety of data files that span many of the disciplines that teach statistics and a symbol substitution table that allows instructors to match symbols used in the CBE lessons with the ones seen in the classroom text.

TARGET POPULATION

The time commitment and production cost for CBE development is quite large. Therefore, it is important to learn if the proposed project duplicates software already on the market. Equally important is to estimate the size of the target audience that may use the materials once created. CBE materials approach cost-effectiveness when the user numbers are large and the content is stable and not in need of annual updating.

The number of potential users for the statistics package is quite large since statistics is taught and used in so many disciplines. However, for that same reason, it is difficult to identify the target population. Beginning statistics students have a wide range of mathematical preparation. Students may take a statistic course in high school, college, graduate school, or in a professional setting, therefore bringing varied personal experiences to the course. Different disciplines choose to emphasize different types of statistical testing for applications to their fields. The number of statistics textbooks used today is just one example of the diversity of statistics teaching and emphasis.

The statistics course offers several levels of feedback to students when they are responding to questions. Help sequences to explain new equations in fine detail are made available to students who may wish a fuller explanation of unfamiliar mathematical concepts. Many exercises allow the student to repeat the problem several times, altering parameters on each new try, so that the student can see the effect of each parameter on the graph or equation.

SELF-PACED

A traditional statistics course at a college typically consists of large lecture hall groups, meeting several times a week, with somewhat smaller problem-solving sessions, meeting perhaps once a week. The large group may not afford individual students the opportunity to ask their specific questions or to practice a concept fully. The subject area of statistics lends itself to self-paced individualized instruction. Each module contains tutorial instruction and exercises on a new concept. The students can choose to see the entire tutorial section or to try just the exercises in any module from the index matrix. The student can see the material in these different formats because these options were carefully planned for and included in the overall design of the course. Each student now has a patient tutor to help them struggle through difficult concepts or obscure equations at the pace they set.

ADAPTABILITY

Statistics is a large subject matter area with a diverse target population, therefore it is important to build flexibility into a computer-based statistics package. CBE has the ability to handle diversity. Students can be given options based on their mathematical abilities. Examples can be varied to relate to different disciplines. Several levels of instructional help and feedbacks to response can be incorporated into the CBE materials to help students at various levels of understanding. But most CBE materials are not flexible and do not provide intelligent help or feedback to struggling students. Even though it is possible to provide flexibility, such a lesson requires good planning, imaginative design, and sophisticated programming - and these components demand a great time commitment.

The University of Delaware has years of experience with students using CBE materials and the two most often heard complaints about the lessons are the frustration of being trapped in a lesson with no way out and the pickiness of the answer judging. In the statistics course, a library of routines is used by each module that allows for complete user control of flow at all points in the lessons. The user can review a previous display, return to the index, ask for a glossary term, leave the lesson, or several other options at any point within the lesson. The library routine creates a consistency of keypresses and a consistency of options for the user throughout the package. To address the answer judging problems, standards for acceptable answers were devised and incorporated into the design of each question. Explicit directions for expected answer formats are given at each question. This solution is not as automatic as the control problem but if carefully implemented can be successful.

STABLE CONTENT

The development time for computer-based course is extensive. It is not cost effective to produce a course therefore if the content is changing so rapidly that it is impossible to keep the material up to date. In addition it is a good design feature to exclude examples from the course that may date it, such as reference to current events or leaders. The statistics field on the whole is unchanging, but new topics especially in exploratory data analysis and regression are always being introduced. In the statistics course, new topics were included together with the basic introductory material. When data files were selected, its timeliness was a large factor in deciding whether or not to include a particular file.

GRAPHICAL APPLICATIONS

Statistics is highly mathematical and abstract at times. The graphical capabilities of the computer-based environment are particularly useful when illustrating the graphical interpretation of an equation or the distribution of a certain function. It was quite easy to split the screen and display the graph at one side of the screen and the equation and derivation beside it.

RECORD KEEPING

A student router allows the instructor to order assigned lessons to various groups of students, permitting the creation of an individualized assignment list for each student group. Certain lessons can be restricted until the appropriate time in the course when that topic is covered. The router also stores a large amount of information on each student and each student group. Signon times and data are available, as well as completion and final grade data for each student for each lesson. These data are available in several formats -- tables, cross-tabulations, and histogram output.

The student router is built on top of the student routing system which is part of the Courseware Authoring System (C.A.S.), Digital's Computer-Based Educational software. Students are registered by batch group assignment, and each group has an individualized group assignment list of lessons. The instructor can specify what data to keep on individual's and group's progress for assigned lessons. Students choose to do assigned lessons, to browse through other lessons, or to read mail.

CONCLUSION

Stepwise needs analysis combined with imagination can yield innovative new designs for courseware. The Courseware Authoring System, with its student router, record keeping, and CBE language, DAL, provides a good environment for producing high quality CBE courseware. A careful needs analysis and a powerful set of development tools enabled Delaware to meet its goals of producing a user-oriented, well-planned, and flexible statistics course.

THEORY VS REALITY
APPLICATION AND MODIFICATION OF INSTRUCTIONAL DESIGN TO TRAINING SITUATIONS

Richard Reinhart
Microtel Learning Services

ABSTRACT:

This presentation will emphasize the practical realities of applying Instructional Design models to actual training situations. All too often the situation dictates alterations to the classic ID models. Training application situations will be compared and contrasted with classical design intentions. The emphasis will be placed on practical differences and where they'll most likely occur.

Many new Instructional Designers, fresh from training, plunge into the profession only to discover many steps in the classical Instructional Systems Design (ISD) process are discarded or grossly violated. One can either suffer the frustration of applying pure ISD or adjust to the realities of practical application.

Generally, a training program can be divided into two categories: in-house projects or outside contract projects. An in-house training program often has the dubious advantage of the training developer as acceptor (the person who must judge the quality of the training project and "accept" it). In other words, the only people they must satisfy are themselves. Not even the most ethical purists could disagree that in many cases, the "pressure" to perform is somewhat reduced. An in-house project may present an opportunity to practice pure ISD if time and resources allow. It usually does not have as many restrictions as an outside contract job.

An outside contract job basically means dealing with and "pleasing" someone outside of your immediate organization. This could range from another department inside your company to contractual obligations outside your company for profit purposes. It usually means a transfer of funds. One of the big departures from the classical ISD model is that of customer approval. An outside contract training program usually requires a lot of customer approval throughout the instructional development phases. This sets up a situation where it is not a question of good lesson objectives, media selection or most suitable CBT system etc., but does the "customer" agree with your selection.

Customer approval can severely affect media selection, course objectives, course length and other instructional ingredients all affecting quality. Depending on the organizational hierarchy, this approval process may slow the project down considerably if there is an outside approval process every step of the way. Should we try to eliminate the customer approval process to bring the world's of theory and practical application closer? No, this process is necessary for the training developer's own protection. If halfway through the project development the customer decides to change their mind about course content, the developer may be stuck with the expense of the analysis changes and corresponding redevelopment. As long as there is a step by step approval/sign off procedure, the customer would be responsible for expenses in changes to items they already approved of.

In standard ISD procedures, we have been taught to select the media after the objectives are developed in the analysis phase. In the real world, however, many projects are initiated with the instructional media already selected or severely restricted choices. This is often because the budget is the major consideration, particularly for outside contract training programs. Most trainers are faced with the problems of low budgets where the appropriate media is not affordable. However some trainers (most likely those in dedicated mainframe CBT) are faced with the uncommon problem of (believe it or not)too high a budget. If there is a huge cash outlay for a large training system (ie CBT), there is a certain pressure to "utilize" it as much as possible. Certain institutions practice a "use it or lose it" policy of budget allocation so that they are encouraged to buy fancy training systems which are like "new toys" they feel compelled to use. This may result in forcing instruction to fit the media even if it is not suitable.

This has been the case for several CBT projects which results in poorer instruction in the form of the classic "electronic page turning" that gives all CBT a bad name. Budget may also restrain the time of development which often translates into reduced quality as well as unsuitable media selection. Unfortunately, restricting quality control often has a more profound effect in CBT than non-CBT media. This is because CBT courses allows a larger student-instructor ratio than non-CBT and as a result tends to be more of a "stand-alone" instructional delivery. Courses that have a higher degree of instructor interaction have a much better chance (if the instructor is good) of adjusting for poor quality courseware. Organizations that rely on CBT courseware suffer the effects of poor quality far more.

Another important element of real world instructional design application that is often left out of the model is that of time constraints.

Missing deadlines for an inhouse project may result in problems for the performance evaluations of those involved but such a problem with an outside contract job may result in monetary penalties or even cancellations. Therefore, time constraints may force developers to rush out the product, select a less suitable media with a faster development time, skip ongoing student evaluations etc., all a compromise of quality. Contract training may also place unusual restrictions on the pure instructional design model. An example is the customer or an outside source supply who the developer may be obliged to use despite its quality or format.

In summation, many people leave school armed and eager with their instructional design theory only to be frustrated in application. One way to avoid this is to stay away from these jobs in the first place (this is often hard to determine in a job interview). Another suggestion is to try to adjust and live within the "system". This may often result in a compromise of your "hard-earned" values. Remember, that instructional design theory is just that, theory. It is a guideline for situations of optimum conditions. The challenge to training professionals is to incorporate it into reality rather than reality into it. The challenge to higher education is to prepare training professionals for realistic situations as well as optimum conditions.

THE IMPLEMENTATION OF A CBT COURSE FOR A SOFTWARE PACKAGE THAT MONITORS THE PERFORMANCE OF AN OPERATING SYSTEM

TANIS M. STEWART - CHEVRON CORPORATION

EMILE E. ATTALA - CALIFORNIA POLYTECHNIC STATE UNIVERSITY

ABSTRACT

This paper summarizes the effort spent to develop and implement a CBT course for a software package that monitors the performance of the MVS-XA operating system used at Chevron Corporation. This software package, Omegamon, is used on the IBM 3081, at both West Coast data centers.

Omegamon was developed by the Candle Corporation to be used in two complementary ways. In system programming it can run as two separate applications providing over 500 commands to display the internal workings of MVS. In operations it provides automatic exception analysis used to check and pinpoint potential troublespots. The purpose of this project was to provide training for those working with the operations functions of Omegamon. The focus was to teach operators how to use the various tools provided by Omegamon. This included exposure to all of the commands and output that are generated when the Omegamon screens are viewed.

OMEGAMON

A course, "Introduction to Omegamon", was developed using IBM's Interactive Instructional Authoring System (IIAS) and implemented using the Interactive Instructional Presentation System (IIPS). IIPS is an online training system that provides for the delivery of computer based courseware via an IBM mainframe terminal network. The IIAS provides the capabilities for in-house courseware development.

There was a strong need for operators to have hands-on training with the software package that monitors the performance of the operating system. But, because the Omegamon screens change so quickly as it monitors the system, it is difficult to provide predictable, consistent hands-on examples in a classroom or on the job training setting. Therefore, incorporating a manual in a computer assisted instruction environment was the answer to this training need. As Omegamon monitors all tasks on an operating system as well as itself, it basically provides three major functions: Timely notification that a problem exists; Problem determination information to assist in pinpointing the areas causing the problem; Information to distinguish which jobs are causing the problem, which are being hurt by the problem, and to what extent they are affected.

The many aspects of the MVS-XA operating system are controlled by thresholds set by operations. Omegamon tells how these are set, when they are being pushed or overrun, and what the system is doing to handle such problems.

Central to this monitoring function is a feature called Exception Analysis Facility. Rather than simply producing displays which the user must examine and interpret, exception analyzers produce a message only when a parameter goes above or below a limit, or when some unusual condition occurs unexpectedly.

The main function of Omegamon is to inform and it does this through a collection of CRT screens. The screens are the heart of Omegamon. There are two kinds of screens used at Chevron, PF-driven and Command screens. The PF-driven screens are vendor supplied, standard function for Omegamon. They are accessed through PF keys. The Command screens were designed by Chevron for

BEST COPY AVAILABLE

use in monitoring specific areas. They are accessed by various command names. These screens are illustrated in figure 1.

The screens are comprised of commands and output. The commands are usually abbreviations for parts of the operating system. The output that appears for them gives statistical information about the specific areas they identify.

Most of the commands on the screen are in place already; the operator does not have to enter them, but does have to interpret them.

To introduce Omegamon and its capabilities, a reference manual was created.

REFERENCE MANUAL

The reference manual was developed to serve a dual function. It is a quick reference guide for operators on-the-job, and it is supporting the student material for the introductory computer based course. Therefore, its structure and content were dictated by these functions.

It was found that to understand Omegamon/MVS-XA, one must have a general understanding of the MVS-XA operating system. With this knowledge many Omegamon features are self-explanatory as soon as the commands are deciphered. Thus, the manual sketches an outline of these MVS-XA concepts and components. It covers them in broad terms with an excellent definition of each.

To the operator, Omegamon is a collection of screens. Therefore, it was imperative that the screens and their content be explained in great detail. The majority of the manual is devoted to coverage of the screens. Since each screen has a different monitoring function with different commands, each screen is identified and explained, individually.

The screens are divided into sections information, output area etc., which are comprised of various major and minor commands and symbols. All of these commands and symbols convey to the operator the status of jobs on the system. The manual states the purpose of each screen, shows the screen in its entirety and then dissects the sections and individually describes or defines each command and symbol.

The other critical point is the area that handles Exception Messages. Operators must be able to recognize and interpret exception messages and take the appropriate action to correct the problem. The manual thoroughly covers this area by showing sample exception messages grouped according to relevent MVS-XA component. Below each message it presents an explanation and the action to correct it.

COMPUTER ASSISTED COURSE

The purpose of our effort was to put the reference manual into a CBT mode of instruction providing the student with tutorial and drill and practice units. Mastery of the material was an important consideration thus the design of the course is cumulative. Each session consists of text screens which introduce and discuss a main concept, and assign a corresponding section in the manual. This step is repeated as many times as is needed to cover the session material. Each session ends with a quiz to measure comprehension.

The methodology used for progressing through the course is shown graphically in figure 2. If students score less than eighty-percent they are placed at the beginning of the quiz for a tutorial and a review of the reading material before they retake the quiz. If students score eighty or higher they branch to a tutorial which reviews the questions they answered incorrectly. Students who score one-hundred percent are automatically branched to the beginning of the next session.

This methodology was chosen because it provides the best learning opportunity. Due to the technical nature of the subject it is extremely important that students have a hard copy to use for reference, however, it is equally important for them to be exposed to Omegamon as it appears on their CRT screen. This design allows students ample opportunity to learn all of the material and review it when necessary. When operators complete the course they not only understand the entire manual but also have an excellent job aid. The text and quiz questions focus on the areas that are essential for using Omegamon. To give hands on experience, simulations of the exception messages are used. Students are asked to identify the problem or the action needed to solve the problem. By utilizing this method operators can be exposed to all system problems that can occur.

INTERACTIVE INSTRUCTIONAL PRESENTATION SYSTEM

The course was created on IBM's Interactive Instructional Authoring System utilizing the course structuring facility authoring language. It was

BEST COPY AVAILABLE

implemented on the Interactive Instructional Presentation System.

Standard screen skeletons reside on an authoring dataset within Chevron. These screens were originally created using the course structuring facility and are used when authoring IIPS courses. These skeletons are copied into a file and used to create the framework for all courses. Included are a set of standard screens used for all Chevron courses as well as text screens and various question screens used to formulate quizzes.

IIPS guides students through the course by giving instructions as necessary. The system also has an automatic tracking mechanism that allows students to sign on and off at any time, but always returns them to the point they left. In addition, it allows for simulations of applications. This function was used to simulate the exception messages for one portion of the course.

SUMMARY AND CONCLUSION

A course was designed, developed and implemented on the IBM 3081 to train computer operators to use a software package, Omegamon, that monitors the performance of the MVS-XA operating system used at Chevron Corporation. The courseware was developed in conjunction with a reference manual that allowed the operators to freeze an Omegamon screen to be able to digest its content. Drill-and-practice examples were created to reinforce the content of each screen.

The Interactive Instructional Authoring System was used to create the courseware together with a set of skeleton frames developed at Chevron to enable the courseware designer to produce cost-effective instructional material. This course is an introduction to the subject area. Future plans at Chevron are to take advantage of the new technologies that are being added to a CBT system and to transfer more of the content of the reference manual into CBT. The challenge in the new design would be to orchestrate audio visual tools to bring alive to the computer operators the functions that each Omegamon screen conveys to them.

REFERENCES

CANDLE CORPORATION (1983). Omegamon Reference Manual, Document number OM53-0175-6.

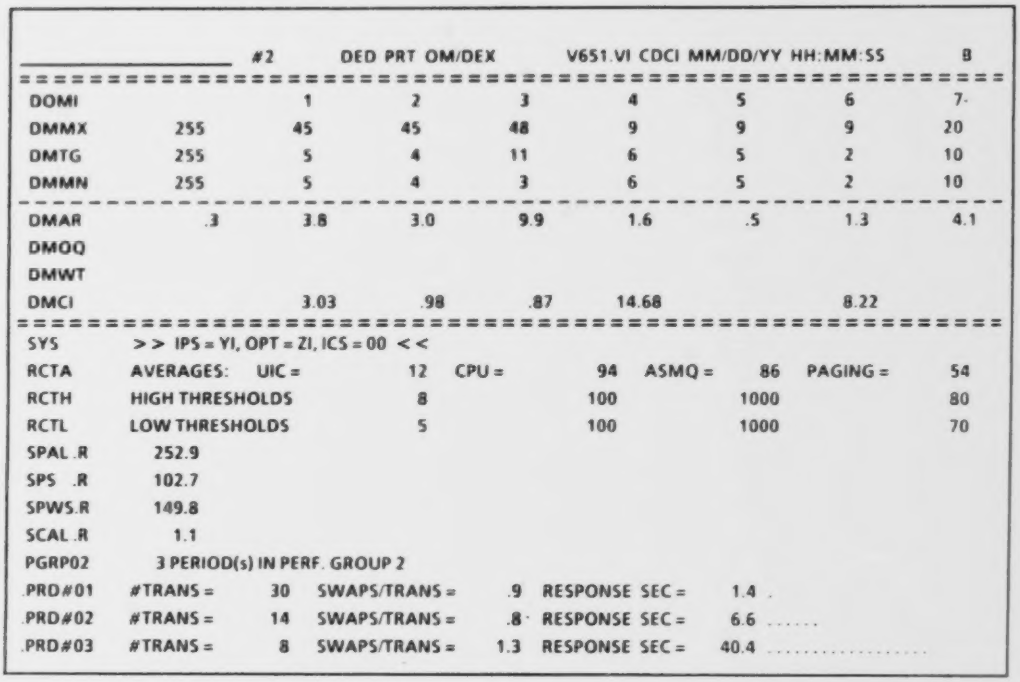
Chevron Corporation, Computer Services Department Training Section (1984). Introduction to Omegamon.

IBM Corporation (1984). Course Authoring Guide, Document Number SH20,0019-0.

IBM Corporation (1980). MVS/OS/MS2 Overview.

IBM Corporation (1981). MVS/SP General Information, Manual, Publication Number GC28-1025.

PF DRIVEN

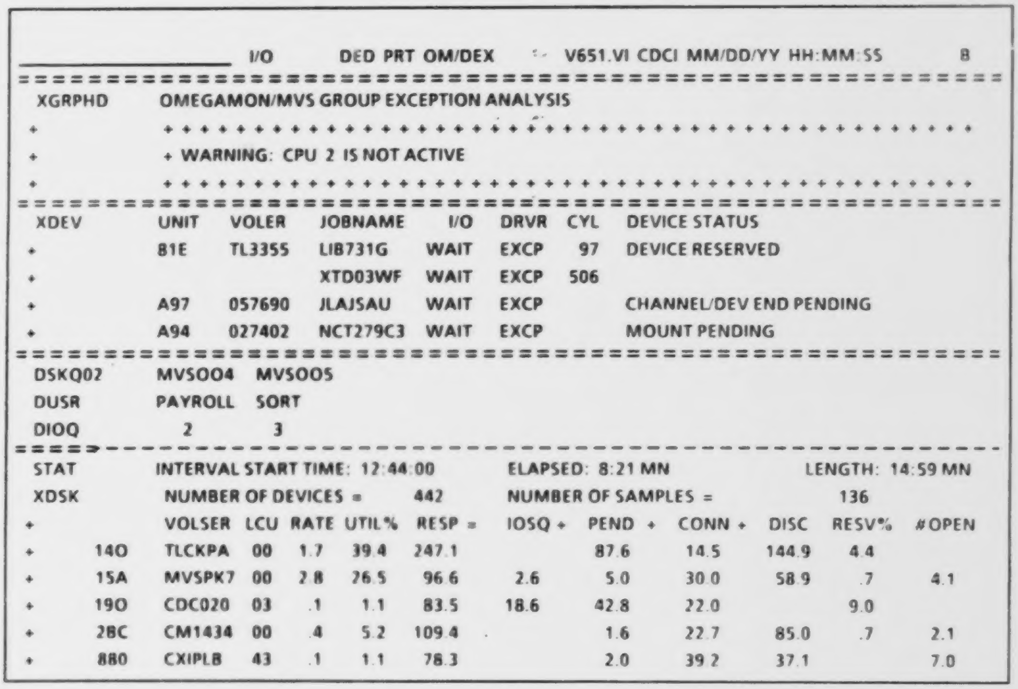


```
_______________   #2      DED PRT OM/DEX        V651.VI CDCI MM/DD/YY HH:MM:SS        B
========================================================================================
DOMI                        1        2        3         4         5         6        7
DMMX          255          45       45       48         9         9         9       20
DMTG          255           5        4       11         6         5         2       10
DMMN          255           5        4        3         6         5         2       10
----------------------------------------------------------------------------------------
DMAR           .3         3.8      3.0      9.9       1.6        .5       1.3      4.1
DMOQ
DMWT
DMCI                     3.03      .98      .87     14.68                 8.22
========================================================================================
SYS       >> IPS = YI, OPT = ZI, ICS = 00 <<
RCTA      AVERAGES:   UIC =         12   CPU =        94   ASMQ =     86   PAGING =     54
RCTH      HIGH THRESHOLDS            8               100            1000                80
RCTL      LOW THRESHOLDS             5               100            1000                70
SPAL .R    252.9
SPS  .R    102.7
SPWS.R     149.8
SCAL .R      1.1
PGRP02     3 PERIOD(s) IN PERF. GROUP 2
.PRD#01   #TRANS =      30   SWAPS/TRANS =    .9   RESPONSE SEC =    1.4 .
.PRD#02   #TRANS =      14   SWAPS/TRANS =    .8·  RESPONSE SEC =    6.6 ......
.PRD#03   #TRANS =       8   SWAPS/TRANS =   1.3   RESPONSE SEC =   40.4 ..................
```

FIGURE 1

OMEGAMON SCREEN

COMMAND DRIVEN



```
_______________   I/O      DED PRT OM/DEX      V651.VI CDCI MM/DD/YY HH:MM:SS        B
========================================================================================
XGRPHD      OMEGAMON/MVS GROUP EXCEPTION ANALYSIS
+           + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +
+           + WARNING:  CPU 2 IS NOT ACTIVE
+           + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +
========================================================================================
XDEV        UNIT   VOLER    JOBNAME    I/O    DRVR   CYL   DEVICE STATUS
+           81E    TL3355   LIB731G    WAIT   EXCP    97   DEVICE RESERVED
+                           XTD03WF    WAIT   EXCP   506
+           A97    057690   JLAJSAU    WAIT   EXCP         CHANNEL/DEV END PENDING
+           A94    027402   NCT279C3   WAIT   EXCP         MOUNT PENDING
========================================================================================
DSKQ02      MVSOO4   MVSOO5
DUSR        PAYROLL  SORT
DIOQ           2        3
====>-----------------------------------------------------------------------------------
STAT        INTERVAL START TIME:  12:44:00      ELAPSED:  8:21 MN        LENGTH:  14:59 MN
XDSK         NUMBER OF DEVICES =      442       NUMBER OF SAMPLES =           136
+            VOLSER  LCU  RATE  UTIL%  RESP =   IOSQ +  PEND +  CONN +  DISC   RESV%  #OPEN
+      14O   TLCKPA  00    1.7   39.4   247.1            87.6     14.5   144.9    4.4
+      15A   MVSPK7  00    2.8   26.5    96.6    2.6      5.0     30.0    58.9     .7     4.1
+      19O   CDC020  03     .1    1.1    83.5   18.6     42.8     22.0             9.0
+      2BC   CM1434  00     .4    5.2   109.4 .           1.6     22.7    85.0     .7     2.1
+      880   CXIPLB  43     .1    1.1    78.3             2.0     39.2    37.1            7.0
```

FIGURE 1

OMEGAMON SCREEN

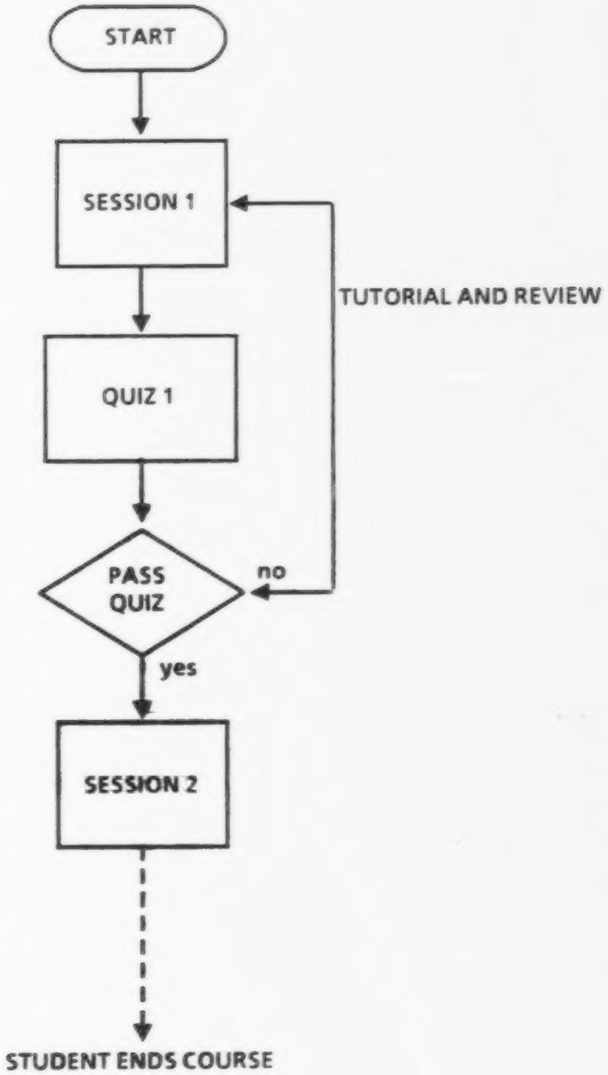

START
SESSION 1
TUTORIAL AND REVIEW
QUIZ 1
PASS QUIZ
no
yes
SESSION 2
STUDENT ENDS COURSE

FIGURE 2

BLOCK DIAGRAM OF A LESSON

Intelligent Tutoring Systems: An Explanation

Kathleen M. Swigger- NTSU

Steven F. Brown - NTSU

Abstract

This paper discusses the nature of Intelligent Tutoring Systems and their various components. In writing this paper the authors have tried to answer such questions as: Why are these systems considered "intelligent"? How do you design one? What do I need to know to build one? The paper also discusses some of the problems confronting designers of intelligent tutoring systems and questions that remain to be answered.

During the past year, the words "Artificial Intelligence" have gained increasing attention from everyone in the computer industry. The recent success of expert systems has made AI technology both practical and profitable. This growing awareness of AI and AI programming techniques has impacted the educational world in the form of computerized tutoring systems (sometimes referred to as Intelligent Computer Assisted Instruction). These systems, which include SOPHIE (Brown et al., 1982), WEST (Burton & Brown, 1982), GUIDON (Clancey, 1982), WHUMPUS (Goldstein, 1982), Meno-II (Soloway et al., 1981), provide problem-solving and tutoring capabilities in such areas as debugging electronic circuits, medical diagnosis, a game that draws upon the student's knowledge of arithmetic, and a system that gives practice in Pascal. Yet, in spite of these gains, designing ICAI systems remains somewhat of a mystery to the majority of people in computer-based education. It is the purpose of this paper, therefore, to try and explain the difference between more traditional CAI and Intelligent CAI and also give readers some ideas of how to design intelligent tutoring systems.

The course designer of conventional CAI envisions any lesson as a series of logical frames or questions, each with its separate answers and responses. Although the course may be implemented eventually as a series of procedure calls or using an authoring system, the author views the logical construct of the course as a series of frames. This is why conventional CAI is sometimes referred to as Frame-Based CAI (Carbonell, 1970). A more traditional CAI environment does not possess any "knowledge" of the subject matter nor does it possess a model of the student it tutors. Conventional CAI systems are programmed not only to teach the same way to all students but also to teach the same way every time they tutor. Although designers of CAI systems may incorporate "remedial" branching for students, they are not able to provide a truely flexible environment.

In contrast to the above approach, an ICAI author thinks of the same lesson as a mix of expert knowledge, a student model, some appropriate teaching strategies, and a communication system. Developers of ICAI have concentrated their efforts in these areas to produce a tutor which knows its subject, knows the student, knows how to become a better teacher, and knows how to communicate with the student. Most literature on ICAI refers to these domains as separate modules to allow for ease of discussion and explanation (Roberts & Park, 1983; Swigger, 1984). In practice, however, the modules may not be well-defined. Following is a summary of these modules and how they are designed.

THE EXPERT MODULE

The expert module contains the

knowledge about the domain the system is trying to teach. This knowledge includes both the facts to be taught and the procedures used by experts to solve the particular problems. For example, if the author is trying to teach the student how to balance a checkbook, then the "expert" module would contain facts about checks (What is a check?), debits (What is a debit and why do I need to know about debits?), as well as the procedures used to compute debits and credits.

One of the most difficult parts of designing an ICAI module is to determine how to represent or store these facts and procedures so that they can be easily maintained and retrieved. The major function of this module, and the reason why they are called "intelligent" is that the system can deduce new facts without having to store everything explicitly in the data base. Various methods and data structures have been proposed to achieve this particular goal.

One of the more popular representations is that of production rules which can be used to represent knowledge about the domain. Production systems usually consist of three parts, a rule base full of rules and facts, a context which is a list of known or initial attributes, and an interpreter which determines which rule to activate next. A classic example of how such a system works is:

English version
P1: If X is a Man then X is Mortal
P2: John is a man.

Logic version
P1: isa(man,X) -→ isa(mortal,X)
P2: isa(man,John).

This system is now able to answer such questions as "Is John Mortal?" isa (mortal,John). The question, isa(mortal, John), is placed in the problem's initial context. The interpreter goes through each of the rules to find a match. On the first pass, isa(mortal,John) matches the THEN portion of P1, isa(mortal,X). The X is bound to John and the right hand side of the rule, isa(man,X), is placed in the context. Since X is bound to John, the item placed in the context is actually isa(man,John). The interpreter then starts the process over and this time matches P2. Since the context and the fact are the same, the system returns true and the inquiry is complete. In such a manner, the production system deduces that John is mortal without having to store the fact isa(mortal,John) in its data base.

In a similar manner, other data structures do the same thing. Winston and Horn (1984) describe a data structure called "frames" which enable them to relate pieces of knowledge through the use of property lists. Semantic nets are also used to derive new facts from lists of related items. Whatever knowledge representation is selected, the major function of each type of representation is to deduce facts without having to represent everything explicitly in the data base.

In addition to facts, the expert module must also know how to use the knowledge. The expert in any field is able to recognize a situation and act accordingly. This procedural knowledge, or inference capability is sometimes a covert solution to a mathematical problem (Blaine & Smith, 1977) or the knowledge of strategies for solving a problem (Miller, 1982). The system may also call upon its procedural knowledge to explain the steps used in an algorithm (Sleeman, 1984).

Equipped with both facts and procedural knowledge, the expert module can intelligently generate answers concerning why a particular action was taken or how a rule was used in the solution of a problem. Swartout's (1983) XPLAIN is an excellent example of the application of an expert system to facilitate an explanation capability.

THE STUDENT MODULE

A model of the student's understanding of the instructional material is contained within the student module. The student module models the behavior of each student through a number of techniques which range from simple storage of topics completed through analysis of students' misconceptions. The model may contain a storage of facts the student knows related to the topic, rules the student has used, or even teaching strategies which have produced favorable results in past tutorial sessions.

According to Carr, Brian and Goldstein (1977), ICAI systems have four methods of supplying information about the student to the student module. The first method is through direct conversation with the student. The second method is through an observation of learner behavior during situations requiring the application of learned rules. The third method is termed structural, which is a rough measure of how the student relates one "chunk" of knowledge to another "chunk". The fourth method is through an overall evaluation of the student's past performance which provides a filtered predictor of future behavior.

Some ICAI systems construct the student model by searching for differences between the student's knowledge base and that of the expert. In brief, discrepancy models (Sleeman & Hendley, 1982) infer a student's weakness from observed behavior. Another method of constructing a student model is called the overlay method (Goldstein, 1982) and assumes that a student's knowledge is simply a subset of the expert's knowledge. If the expert's knowledge is represented as a graph with each node in the graph signifying a particular skill, then the student's model is that portion of the graph (skill) that has been mastered. If the student is deficient in any particular skill, then he is given more material of the same skill level or is allowed to approach the problem using different subskill paths.

Another approach to the student model assumes that a student may possess reasoning strategies that are totally different from the expert. When the student performs poorly, it is the responsibility of the student module to diagnose the underlying cause(s) of the errors. O'Shea's (1977) introductory algebra tutor attempts to determine incorrect rules used by the student. One problem arising from this process is that some students may be plagued by so many "mal-rules" (Sleeman, 1984) that isolation may be difficult to achieve. An active area of interest within ICAI involves this detection and correction of "buggy", or faulty strategies used by students. Van Lehn's (1980) work with a theory of repairing these buggy strategies is based on the idea that the student may have a good basic knowledge of rules, but has problems in applying the rules.

THE TUTORIAL MODULE

The tutorial strategy module, also called the teaching module, is responsible for changing the student into an expert in the most efficient and effective manner. Once the student model is created and all errors properly identified, the teaching module can either diagnose (Sleeman, 1984; Soloway et al., 1981), coach (Brown & Burton, 1978), or allow the student to engage in "guided" discovery (Miller, 1982). All three of these methods have been used with some degree of success by authors of various ICAI systems.

Since students like to receive explanations of complex phenomena in a variety of ways, the ICAI systems have the capacity of presenting many possible modes of explanation. STEAMER (Stevens, 1981) was able to shift into nine different modes of explanation, such as explaining how the parts of the power plant fit together, demonstrating cause-effect relationships, and using straight didactic exposition.

Although it seems as if the incorporation of nine methods of explanation would require nine times the amount of programming of the expert module, the explanation facility can be designed in such a way that all methods of explanation draw on the same base knowledge. NEOMYCIN, the infectious disease diagnosis tutorial system (Clancey, 1984), has the ability to generate explanations through the use of a domain-independent explanation-generator.

The tutorial strategy module should also be capable of improving its instrutional ability with time. The previously mentioned algebra tutor of O'Shea (1977) made experimental changes in tutorial strategy and analyzed the results of the changes so that successful strategy changes could be made a permanent part of the module.

As has been shown, there are many approaches to the solution of the problem of making the tutorial strategy module behave more like a teacher than a repeater of information contained in a book. The wide variety of approaches taken assures ICAI researchers that a uniform methodology for the module is still in a nascent state of development.

THE COMMUNICATIONS MODULE

The communications module is limited to generating grammatically correct and rhetorically effective output to (1) help the tutoring component do its job and (2) to interpret the student's responses in terms of the categorizations that the tutoring component is sensitive to. Winograd (1977) notes that the ICAI system needs a communication module capable of complete interaction between expert, student, and tutorial strategy modules. This communication must include the ability to understand the student's responses and respond to the student in a way that the student will understand what is said. GUIDON (Clancey, 1984), the intelligent tutor for diagnosing infectious diseases, has a communications module with the ability to refer to three types of knowledge to assist in an explanation. GUIDON can access data and explain why an expert decision was made, make an abstraction from highly specific knowledge to explain patterns, and, of course, supply purely factual information.

In order to understand student input, be it a question or an answer, the communications module is the vital link between human and some type of data storage. Ideally, the communications module should be able to understand the utterance of the student. This is accomplished by either attempting to parse a natural language sentence or through keyword matches. Traditionally, the communications module is the last to receive the attention of AI researchers, and today reflects the fewest additions of heuristic knowledge techniques. Such research questions remain a fruitful area of concern.

CONCLUSION

ICAI is making and will continue to make major contributions to our understanding of the instructional process. However, one of the major problems facing ICAI designers is the labor intensive development costs associated with these types of lessons. Although the development of conventional CAI continues to be costly, developmental costs for ICAI tend to be even higher and more time consuming. In order to develop a single ICAI module, one must document and describe the reasoning processes of experts and students in the problem domain. This knowledge and knowledge about strategies must then be encoded and represented in the computer.

Another major research focus is on defining teaching and learning strategies. By capturing students' misconceptions or bugs, researchers are developing some ideas of how various knowledge structures are stored inside the brain. It is in this area that ICAI holds the most promise. As designers of ICAI uncover general problem solving rules, the entire education community will benefit from such research.

REFERENCES

Blaine, Lee and Smith, Robert L. Intelligent CAI: the role of the curriculum in suggesting computational models of reasoning. Proceedings of the Association for Computing Machinery, 1977, 241-246.

Brown, J.S., Burton, R.R., and de Kleer, J. Pedagogical, Natural Language, and Knowledge Engineering Techniques in SOPHIE. Intelligent Tutoring Systems, Sleeman, D. and Brown, J.S., eds., Academic Press, New York, 1982.

Burton, R.S., and Brown, J.S., An investigation of computer coaching for informal learning activities. Intelligent Tutoring Systems, Sleeman, D.S. and Brown, J.S. eds., Academic Press, New York, 1982.

Carr, Brian and Goldstein, Ira P., Overlays: A theory of madelling for computer aided instruction. Artificial Intelligence Memo 406, Massachusetts Institute of Technology Artificial Intelligence Laboratory, 1977.

Clancey, William J. Classification problem solving. Proceedings of the National Conference on Artificial Intelligence, Austin, Texas, August, 1984.

Clancey, William J. Tutoring Rules for Guiding a Case Method Dialog. Intelligent Tutoring Systems, Sleeman, D., and Brown, J. eds., Academic Press, New York, 1982.

Goldstein, Ira P. The genetic graph: a representation for the evolution of procedural knowledge. Intelligent Tutoring Systems, eds., Academic Press, New York, 1982.

Miller, Mark L. A structured planning and debugging environment for elementary programming. Intelligent Tutoring Systems, Sleeman, D. and Brown, J.S., eds., Academic Press, New York, 1982.

O'Shea, Timothy. Self-Imrpoving teaching systems, Doctoral Dissertation, University of Texas, 1979.

Roberts, Franklin C., and Park, Ok-choon. Intelligent computer-assisted instruction: an explanation and overview. Educational Technology, Dec., 1983.

Sleeman, D. and Hendley, R.J. ACE: a system which analyzes complex explanations. Intelligent Tutoring Systems, Sleeman, D. and Brown, J.S., eds., Academic Press, New York, 1982.

Sleeman, D. Mis-generalization: an explanation of observed mal-rules, Proceedings of the Sixth Annual Conference of the Cognitive Science Society, 1984.

Stevens, A. STEAMER: Advanced Computer-Aided Instruction in Propulsion Engineering. Report no. 4702, Bolt, Beranek, and Newman Inc., Cambridge, Mass., 1981.

Swartout, William R. XPLAIN: A system for creating and explaining expert consulting programs. Artificial

Intelligence, 21, 1983, 285-325.

Swigger, Kathleen M., Intelligent Tutoring Systems: A tutorial. *AEDS MONITOR*, 23, 1985, 6-11.

Van Lehn, K. and Brown, J.S. Planning nets: a representation for formalizing analogies and semantic models of procedural skills. *Aptitude, Learning, and Instruction- Vol. II: Cognitive Process Analysis of Learning and Problem Solving*, Snow, R.E., Federico, P., and Montague, W.E., eds., Lawrence Erlbaum Associates, Hillsdale, New Jersey, 1980.

Winograd, T. A framework for understanding discourse. *AIM-297*, Stanford University, 1977.

Winston, Patrick H., and Horn, Berthold K.P., *LISP*, Addison-Wesley, Reading, Mass., 1984.

OPERATION OF A MICROCOMPUTER CLASSROOM

USING A NETWORK APPLICATION

Nancy Balogh
Carol Jarom

Office of Computer-Based Instruction
University of Delaware

ABSTRACT

The operation of a microcomputer classroom can be considerably more versatile and better adapted to users' needs when the computers are connected by means of a network application. A network allows each station many of the advantages of a mainframe computer while preserving the versatility of a stand-alone microcomputer. The Office of Computer-Based Instruction at the University of Delaware operates two microcomputer classrooms in the College of Engineering. This paper discusses the attributes of these two networks and shows how these attributes affect the way the classroom is operated.

INTRODUCTION

At the Office of Computer-Based Instruction of the University of Delaware, some of the initial considerations in setting up a network in a microcomputer classroom involved choosing the hardware and software, establishing security procedures, and developing general operational procedures. The goal was to produce a classroom situation in which students could get help during assisted hours, yet have access to much of the software at any time. Another important consideration was gaining faculty involvement in the use and support of the network.

This paper will discuss hardware, software, student use, and security as they relate to the development and operation of the network. Approaches for increasing the input from faculty and students and for evaluating usage of microcomputers will be described.

HARDWARE

Since nearly every software package has specific requirements for hardware configuration, it would be prohibitively expensive to set up every station to accommodate all the software used in the classroom. The attempt to keep costs low by avoiding duplication of expensive equipment has resulted in a different combination of hardware for nearly every computer station within a classroom. For instance, one station is equipped with a modem for communication with the Plato system and other computer systems on campus. Another is equipped with a digitizer tablet and a plotter. Other stations include such equipment as printers, color monitors, and a hardware security device which is required for operation of a scientific word processing package.

SOFTWARE

Certain types of software packages are essential to any microcomputer site--an operating system, a full-featured word processor, and programming compilers. In addition it is necessary to consider the specific needs of the site according to the discipline and the users it is serving.

For the engineering networks, the Office of Computer-Based Instruction (OCBI) provides packages that fulfill the requirements for generalized types of software. In addition, within each department a faculty committee investigates software to fill the department's needs; faculty members who want packages for specific classes make recommendations to this committee.

Each new software package must be treated as a special case in determining network access and signout procedures. Copy-protected software cannot be put on the network, so each user must sign out the disks individually; this limits the use of that software to hours when the classroom has an assistant present. At the other extreme, software packages that support the network application can be used any time the site is open because no disks need to be signed out. Intermediate between these extremes, software that is not copy protected may be placed on the network by restricting its use to correspond to the number of copies purchased.

At the present time, there is a limited availability of software that supports network use, but packages that support the network application offer many advantages, including the following:

* Reduced cost per station
* Increased security
* Increased availability
* Elimination of disk handling
* Ease of program installation

With the increased popularity of networks, many software companies are expected to be marketing network applications of their software in the near future.

The Office of Computer-Based Instruction has attempted to use network versions of software as much as possible. We have been fortunate to be selected as a Beta test site for WordPerfect, a word processor from Satellite Software International. Our users have been extremely pleased with this package; it is the most heavily used software on any of our networks. We have also installed an Ethernet multi-user version of dBASE II which allows four concurrent users and controls access to its files through network semaphores; this too has been well liked by users.

Another vehicle for reducing costs and increasing ease of access and control is the acquisition of site licenses for software. Our version of the Forth programming system, MMS-Forth, is a type of site license referred to as a building license. This allows the program to be used on any IBM PC in the building in which our Engineering networks are located. Thus, in this building Forth is installed on both networks, but computers which are not connected to the network benefit from the license, also. Other types of site licenses can cover a classroom or an entire campus.

STUDENT USE

Currently, a user can obtain access to the network simply by turning on a computer after inserting a network boot disk. This disk is available at all times. Future plans include installation of autobooting ROM chips. When such a chip is installed in the computer it automatically boots to the network unless a disk has been inserted in drive A:.

In order to make the network "user-friendly," a system of batch files has been developed. When the user boots to the network the first file executed, the AUTOEXEC.BAT file, logs the user onto the network, links the printer and the system files, then provides the user a HELP screen briefly describing the software available, as shown in figure 1.

Figure 1. Introductory HELP display.

This information allows the user to select the desired software by typing in a single word. Each word calls a separate batch file which contains commands for linking the necessary network volumes and instructions for beginning execution of the chosen software. An example of the instructions for a user of WordPerfect is shown in figure 2.

Figure 2. Instructions for WordPerfect users.

Students following the above procedures have no need to understand the operation of the network software because all network procedures are handled automatically by the batch files. However, the student who has more complex needs is not restricted. The network software commands are available to all users and can easily be mastered by anyone with limited programming experience.

SECURITY

Although our classrooms are assisted during peak hours of usage, there are many more hours when the classrooms are available for use without an assistant. This makes it necessary to secure all hardware and software to the best of our ability. All items of hardware, including all cables, are locked to their tables through a system of padlocks and brackets. Manuals, disks, templates and assorted pamphlets are kept locked in a cabinet. These materials may be signed out by surrendering University identification only when an assistant is on duty and must be returned when the assisted hours end. Materials may not be removed from the room.

Security of files and signons on the network is provided by controls available within the Ethernet software, including the assignment of file passwords. Volumes can be designated as "public," "private," or "shared"; each designation provides a different combination of limitations on file access. In addition, passwords can be assigned to volumes to further refine the levels of security for files. The combinations of security levels available are shown in figure 3.

Volume Designation	Without Volume Password	With Volume Password	Volume Access
Private	Only the volume owner can use the volume.	Everyone except the owner must give the password.	Read/Write One user at a time
Public	Anyone can use the volume.	Everyone except the owner must give the password.	Read only (except owner) Multiple users at the same time
Shared	Only the volume owner can use the volume.	Everyone except the owner must give the password.	Read/Write Multiple users at the same time

Figure 3. Levels of volume security provided by 3COM Ethernet.

FACULTY AND STUDENT INVOLVEMENT

An OCBI professional is assigned to be the contact person for each department, to keep in touch with department needs, to supervise the operation of the classroom, and to be aware of problems as they arise. "Open House" sessions and demonstrations of software are offered to faculty to increase their familiarity with the network and to answer questions about adapting the PCs and the network to their needs. A newsletter about the network is issued periodically, approximately once a month, to keep users informed about network developments such as the acquisition of new hardware and software, scheduling changes, and innovative uses of the resources by other users.

Faculty who wish to integrate student use of the PCs into their courses may request file space on the hard disk, reserve time for their students on the network, and request training for their classes in the use of the hardware and software needed. Beginning freshmen are now scheduled for introductory explanations of network hardware and software as part of one of their first engineering courses.

EVALUATING FACULTY AND STUDENT USE

It is OCBI's policy to keep accurate records of computer usage. At the present time, usage is being calculated by a head count method during assisted hours. In addition, at each workstation, users are requested to indicate the hours they are working at that station and the purpose of their session. From time to time, users are asked to fill out evaluations in which they may outline their needs and make suggestions for improving the classroom. Future plans include writing software that will keep records of signon-group hours of usage and types of software used.

CONCLUSION

The network has proved to be an efficient and cost-effective medium for delivering software for microcomputers in an educational environment. The increased security of software in network applications and the opportunity for the University community to use the computers at any hour make the network a valuable resource. The potential for future expansion of the network software and hardware provides a degree of flexibility that should allow such installations to adapt successfully to the evolving technology

BEST COPY AVAILABLE

Several Strategies to Maximize the Effectiveness Of Computer Assisted Instruction: Lessons Gained from Observing Student Errors and Preferences

George H. Conklin

North Carolina Central University

ABSTRACT

Programmers and computer professionals enjoy fast terminals and video displays. Unlike professionals, students have different needs. An experiment is described where various versions of the same program were used to teach statistics to undergraduates in a competency-based course in a social science department. Several suggestions are made based on the results obtained from running several different versions of the same program as the minicomputer gave way to the microcomputer and printed output became less fashionable. It is suggested that much can be learned from observing student errors and preferences when computer-assisted instruction lessons are being designed.

INTRODUCTION

This paper describes a seven year effort at teaching statistics to undergraduate social science students with weak math skills using the computer as an instructional tool. The programs, called CASE, (Computer-Assisted Sociological Exercises), have evolved through three generations of computers, having been modified as the result of student response and equipment improvements. Since the course was designed to be competency-based, the program evaluation method used here is the one favored by Patton (1980), where a history of the development process is described and measured. We thus describe how the programs were written and modified as equipment improved and the minicomputer was replaced by the microcomputer. In particular, we try to estimate the effects of how new computers changed the techniques of the lessons, whether the video or printed output was most effective and if lessons with worksheets proved to be more effective than lessons where only math formulas were used.

Molnar (1982) has labeled the use of the computer in higher education a 'new intellectual technology.' According to Molnar, a review of the literature has shown and continues to show that when the computer is used as a learning tool, there is not only a 10 percent gain in learning, but that learning is also up to a third faster. Molnar feels that the consistency of results from earlier studies (Edwards et. al. 1974) and later results (Kulik et. al. 1980) means that the point is solidly established in the literature and needs no further research.

In place of formal studies on whether the computer is as useful as pencil and paper, Molnar argues that research should turn to studying what elements of computer-assisted instruction are most effective. For example, a useful question might be not if computers are effective compared to lectures, but whether using video output might be a better learning technique than relying on printed output. Another important question might be if the speed at which a computer or program can process student input influences learning.

BACKGROUND

Teaching statistics to social science students is probably one of the tasks least enjoyed by professors whose training is not in math. Statistics for such students is difficult to teach because math is often a low priority for social science departments, and the students are often those with the weakest backgrounds in math. At North Carolina Cental University, the site of the experiments described here, students typically take their methods courses in the senior year, having finished only an algebra course in the freshman year.

A particular problem teaching statistics to students with poor math skills turned out to be the problem of algebraic order. When given a simple formula to solve such as how to calculate a standard deviation, students do not remember how to go about substituting numbers into the equation. Other formulas, such as the difference of means test (see Tables I and II), proved even more difficult. It appeared obvious to the author that the computer was a logical method to teach how to solve equations and apply statistics to problems of interest to sociologists. The use of a programmed textbook (Bradley and McClelland 1977) had proven of very limited value because students wrote out the exercises by looking first at the answers.

Computer Assisted Sociological Exercises was designed to meet the needs of a course where every student was to achieve a number of simple goals. For example, it was decided that every student must know how to complete a standard deviation, and be able to compute one on an examination with only very minor errors. Nine lessons covered the basic statistical concepts. So students could get experience with real data, every lesson introduced an interesting problem and showed how data could be used to shed light on the problem. Finally, a formula is given, and the student is stepped through the order which the formula needs to be solved. All lessons contain a homework assignment picked at random from one of over 50 possibilities. The student can use the computer to check his or her answers. If the answers are wrong, a new assignment is given. Competency is the goal, so there is no penalty for missing the answers on the homework.

USING OLD COMPUTER EQUIPMENT

CASE was first written in 1978. At that time, the University of North Carolina provided a centralized facility consisting of two HP 2000 computers for instructional use. A library of instructional packages from CONDUIT was available, all printer oriented. In early 1978 the Apple computer still had integer basic, while the original Pet computers provided maximum flexibility. Since our main goal was to provide equivalent service to students regardless of machine used, CASE was designed to be used with either a 32K Pet with a printer or the HP 2000, accessed with either old and slow ASR33 Teletypes or newer and faster 300 baud Decwriters. Our first experiment was to see 1. which computer or terminal the students favored, 2. whether the speed of the terminal affected student use and 3. whether students had mastered the material to the degree that all could compute standard deviations or similar procedures.

To our surprise, students favored the HP computer, and the Teletype was the terminal which was used first. In fact, students would wait to use the Teletype and pass up the Decwriters or Pet systems. To see why, the author conducted an experiment. Six students were recruited from an adult education class. Three were assigned to the Teletype, and three to the Decwriter. They were given a CASE lesson to do as a group. Since the lessons were self-contained, previous knowledge was not required. On the first round, both teams finished at the same time. On the second round, those with the Decwriter at 300 baud finished only a minute faster. Only on the third round did the team on the faster terminal finish most quickly. A quick look at body language showed why: students on the faster terminal sat away from the terminal until it stopped printing, then would peer forward to see what was printed. Those on the Teletype, however, had their noses glued to the print head and were reading along with the printing. Students thus could read along with the slower printer and favored it. We concluded that faster printers helped those of us programming, but not the students.

As for results on the classroom examinations, the Hawthorne effect prevailed. Seeing how hard the professor was trying, all students mastered the worksheets and calculations for a standard deviation, both for the mid-term and final. This represented a great gain over the book alone. Further, since the output was printed, when cut into sheets CASE was actually very similar to a textbook and could be reviewed for exams just as a book could. Students did not master the difference of means test very well. As the last lesson of the year, the author attributed this to lack of time to learn to mastery (see Tables I and II).

Year two revealed the same patterns as year one, except that the Pets received even less use, and students continued to wait in line to use the Teletype. Achievement on the exams fell about five points as the Hawthorne effect wore off. By year three it was obvious that the lessons on the difference of means was not too effective. Further, it appeared that all other computer lesson writers had shifted entirely to the use of the video screen. Therefore, the lesson on standard deviations was moved to the video screen display on both Pets and several Apple II series computers. What happened?

Students who used the video versions complained that they had nothing to review from. There were problems copying the homework from the screens. Problems included errors in copying numbers, errors in drawing the worksheet shown in Table I, and the general feeling that the Teletype was 'better.' Looking at how the worksheet was copied off the screen, we then decided to provide xeroxed worksheets with the lesson, and to ask students to copy directly onto the sheet the numbers from the screen. This proved more successful, although the Teletype again proved more popular. However, it was the video version which provided a clue as to why the students continued to do so poorly on the difference of means test.

By forcing students to work their way not only through an equation but also through a worksheet, CASE had accidently solved a major problem for students weak in math. Specifically, the worksheet contained within it a procedure for solving the problem of how to calculate a standard deviation. First, students calculate the average, then they can fill out the worksheet in logical order from left to right. On the examination, students wrote down the formulas from memory, but actually solved the equation by memorizing the worksheet. Only by giving out two worksheets, one for practice in the lesson and one for homework was it possible to make the video versions as useful as the printed version. Worksheets partially filled out by the computer turned out to be the reason standard deviations were mastered by nearly all, but the difference of means test were not.

CONCLUSIONS

Patton (1980) has suggested that in the evaluation of educational institutions one must study the history of a problem and how it is solved. He points out that what does not happen is almost as important as what did happen. Because students did not immediately take to the lastest video versions of CASE, and actually seemed to like the older technology, several important lessons were learned. Video screens allow the math formulas to be displayed accurately, but what the students needed were worksheets and how to use worksheets to solve problems. With the old technology, printed output was available for study as homework; video screens were more quickly forgotten. The latest equipment is of great interest to computer professionals, but students had other needs. Fast printing terminals caused students to move away from the terminals as the lesson was printed; slower terminals seemed to increase concentration.

As the CASE lessons move into the seventh year, the author is trying to devise a worksheet for solving difference of means tests. Old technology is fading, and the original HP 2000 computer is being abandoned by the State of North Carolina. The Apple versions of CASE are being redone in high resolution color graphics, but the printers are being kept connected so worksheets can be produced for each student with different homework assignments. Students still use the original Teletype, although this option will be ended when the HP 2000 service is discontinued. The Hawthorne effect has worn off, but competency learning still remains an achievement of the CASE programs. Since students can repeat homework assignments until mastery is obtained, CASE provides a way for an instructor to give out different homework assignments while not having to grade more papers. Using CASE, learning to mastery can be achieved even by students whose knowledge of math formulas is poor. Since our mark of success is complete mastery by every student of specific tasks in the course, CASE has proved to be of mixed success. The lesson on standard deviations is successful, but the effort to teach the difference of means test is much less successful. However, for all lessons, CASE has enabled students to improve learning over and above the textbook level.

The author has informally discussed the results of this paper with professional math teachers. The general response has been that students simply should be taught the rules of algebra or failed out. Further, I have been told that no self-respecting math teacher would teach 'like that.' However, further investigation has shown that in the field of adult education and worker training, it is often assumed tht workers not only do not know the rules of algebra, but that it is not a good use of time to try to teach such rules. A trainer for telephone linemen reported that to teach Ohm's law (E equals IR) a wheel device is used so the students can solve the equation. For a job as complex as solving for voltage (E) when given power consumption in watts, a very complex wheel is memorized by students. Such devices are routinely used when amateur radio courses are taught.

Even using the computer to make certain students are 'walked through' the rules of algebra does not mean that such rules can be remembered long enough for everyone to master every lesson to achievement. Rather, the best use of the computer seems to be to make certain that students master a worksheet to solve a problem. If the goal of the class is to pass some students and to fail others, then the textbook approach will give the professor a curve ranging from A to F. If the goal of the class make certain that every student masters how to do a procedure like a standard deviation, then the computer can be declared a success when the computer is used to teach both algebraic order and how to use a worksheet outlining the steps which must be used to solve a problem.

REFERENCES

Bradley, J. and J. McClelland (1977) BASIC STATISTICAL CONCEPTS. Dallas, TX: Scott, Foresman and Company.

Edwards, J., et. al. (1974) 'Is CAI Effective?' AEDS JOURNAL (Summer):122-126.

Kulik, J., C. Kulik and P. Cohen (1980) 'Effectiveness of Computer-Based College Teaching: A Metaanalysis of Findings.' REVIEW OF EDUCATIONAL RESEARCH 50 (Winter):525-544.

Molnar, A. (1982) 'The Search for New Intellectual Technologies.' T.H.E. JOURNAL (September):104-112.

Patton, M. (1980) QUALITATIVE EVALUATION METHODS. Beverly Hills, CA: Sage Publications.

Table 1

Sample of Homework from CASE Lesson 5, Showing Worksheet Used in Solving a Standard Deviation

The computer will now select for you at random from the states. You should use the output as a worksheet to calculate the standard deviation of the numbers listed. Next week you should use the correction code to see if you worksheet is correct. Your code for corrections for next week is 4.

State	Rate	Rate - Mean	Rate - Mean Squared	Sum of Squares
4	24			
5	22			
6	19			
7	26			
8	22			
9	30			

Step 1. Fill in the mean ( ).
Then using the steps outlined above, fill in the remaining values and calculate the standard deviation.

Table 2

An Example of a Less Successful Use of the Computer: Formulas are Explained, But No Worksheet is Provided

The formula for the difference of means test is:

$$\frac{X - XBAR}{\frac{s}{\sqrt{N-1}}}$$

(The formula is printed after about three pages of explanation and interaction with the student).

Example 1

You know from company records that the average wage for a company you are studying is $8,000. You have talked to 100 randomly selected workers and find that the average of the 100 workers is $7,875 yearly, with a standard deviation of $1,000. Is your sample biased?

Let us apply what we have learned above:

1. We must first establish the numbers we are going to use:
   A. What is the sample size? (answers are processed)....
   B. What is the square root of the sample size (N)?
   C. What is the standard deviation?
   D. Now we must divide the standard deviation of 1000 by the square root of N. What is the answer?

The program continues to step the student through the steps until the answer is obtained. The word problem is not difficult for students on an examination. They just forget the order which must be used to solve the equation. The worksheet in Table I above solves this problem for the standard deviation (and correlations).

DEVELOPING CAI IN PASCAL

Louisa Frank

Office of Computer-Based Instruction
University of Delaware

Abstract

There are many programming languages available for microcomputers. Each language has certain characteristics that, in most cases, are disadvantageous when used for computer aided instructional development. When choosing a language for CAI development, the language under consideration should be well documented, well used, and fully supported by the target machine. It should not limit the lesson design and should be fairly transportable.

The qualities above plus additional advantages all point to a language being used for development at the University of Delaware Office of Computer-Based Instruction - Pascal.

Introduction

Many programming languages are available for use on microcomputers. However, only a few of these languages are primarily suited for computer-based instructional (CAI) development. When choosing a standard microcomputer language, such as BASIC, Pascal, Fortran, or C, to do CAI development, the advantages and disadvantages of the specific language, in terms of CAI, must be taken into consideration. A language should display all of the following qualities:

- A well documented, well used language, fully supported by the target machine, is, in most cases, advantageous to use, rather than one that is new and, perhaps, riddled with unknown errors.
- The limitations of the chosen language should neither hinder innovative or imaginative ideas nor slow or halt development.
- If the lesson is to be transportable, choose a language that will make this task easier to accomplish.
- Programmers skilled in a more common language are more readily available than those that have mastered an uncommon language.
- It is wise to choose a language that will be in existence for some time after the lesson has been completed. Often a program has to be modified after it has been finished.

One language that encompasses all of these specifications is Pascal. Besides being an established, well used language in the mainframe as well as microcomputer field, it has been in existence for many years. It is, in addition to BASIC, one of the more common languages for programming.

Advantages of Pascal

Pascal, specifically, Microsoft (MS) Pascal, was chosen for developing several lessons on the IBM PC at the University of Delaware Office of Computer-Based Instruction. The lessons had a set time line to follow, so a language that demonstrated all of the previously mentioned qualities was a must. The advantages and disadvantages of the language as compared to several other languages available for the IBM PC were discussed at length before any programming took place.

Because Pascal is such an established programming language, it is well documented. Technical errors are usually well known as are solutions to these errors. Many companies that market various Pascal language versions have included, in written documentation, the specific known faults of their version. They also may have a consulting staff on duty to answer technical questions. OCBI found MS Pascal to be well documented. (This may or may not be true of all Pascal versions.) As well documented as it is, though, programmers encountered errors that were unknown even to the company consultants. This disadvantage is a hazard to all programming languages, not just the one OCBI chose, and not just Pascal.

Since many people are familiar with Pascal, consultants and programmers are abundant. The availability of programming staff when creating a CAI lesson is a major consideration. It is much easier to find and hire Pascal programmers than to locate programmers for some uncommon or fairly new language. The language, itself, is not as difficult to learn as other, more complex languages. When the time came to hire programming staff for the lessons, there were several excellent programmers, well versed in Pascal, available. Although these programmers did not have any experience with microcomputers or the specific version of Pascal that OCBI was using, it took relatively little time for them to become acquainted with the IBM PC and the language.

Several versions of Pascal exist for many different machines. Each version is slightly different, but most adhere to the same basic structure. If the lesson does not require machine specific procedures, it is much easier to transport to another machine. Since the languages on both machines are basically the same, the code can be transferred to the new machine. Of course, each Pascal differs so some code must be changed. However, changing part of the code is a much more desirable solution than totally rewriting the entire program in another language.

The basic structure of the language makes programming and debugging an easier task. Code can be broken up into logical procedures, and the programmer can make the code very legible, not only through commenting, but also through indenting and separation. Most Pascal languages also allow the programmer to break the program into segments (usually called modules or units). This creates a more logical program that is easier to modify. The code written at OCBI was broken up into several modules. Each module contained a specific part of the lesson. For instance, one module contained the title page while another held a simulation used later in the lesson. When a modification was needed in a particular part of the lesson, only the specific module that contained the code needed to be altered. To institute the change, only that particular module needed to be recompiled, not the whole program. This saved a considerable amount of time. Figure 1 shows a program, DRIVER.EXE, and its associated modules (*.PAS). The program has been segmented into several separate units, all with the .PAS extension. Figure 2 displays some of the actual code of a sample module.

DRIVER.EXE	This is the full, executable lesson. It is composed of all compiled source files.
DRIVER.PAS GLGTITLE.PAS GLG1P1.PAS GLG2P1.PAS GLG3P1.PAS	These are some of the individual source files. Each is a separate module holding code specific to that part of the lesson. For example, glgtitle holds all of the code for the title page. Glg1p1 contains the code for page 1 of part 1. Each of these modules can be compiled separately.

Figure 1. DRIVER.EXE and its modules.

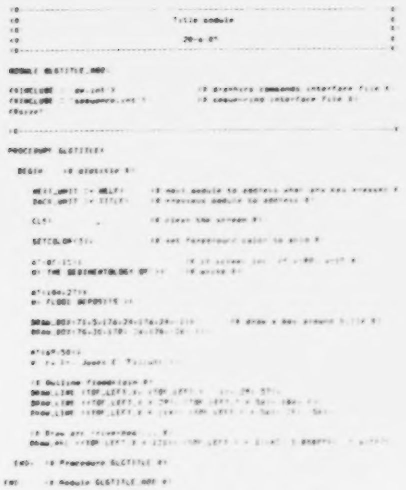


Figure 2. Sample of code from DRIVER.EXE.

Another plus feature is the ability to combine Pascal code with code written in another language. Because most languages are suited to a particular application, in some cases, it may be better to write a routine in another language. Fortran is usually considered when writing engineering applications or complex mathematical functions. With MS Pascal, code written in Fortran can be combined with Pascal code to produce a powerful lesson. Other languages, such as C or Assembler, may be combined with Pascal. One lesson created on the IBM PC had code already written on a mainframe computer in Fortran. This code was downloaded to the IBM PC and incorporated into the overall Pascal lesson. Had the feature of combining code of different languages with Pascal not been available, the complex Fortran code would have had to have been rewritten in Pascal. This would have taken much more time than the Office had been allocated for completion of the lesson.

Another advantage of Pascal over other languages is the availability of programming tools. Many more tools, written to help the programmer simplify the task of writing code, are marketed for Pascal than for other complex languages. This is, again, probably because the language is so well established. Even so, OCBI found that many tools were not available that could have shortened development time. For this reason, specific time was set a side to create several programming utilities. These tools, once written, dramatically decreased the programming time necessary for development.

Disadvantages

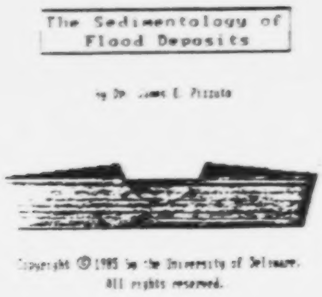
Along with all the advantages of using Pascal versus other programming languages, there are disadvantages. Besides the few already mentioned, a major drawback found when developing at OCBI was that Pascal was not a good language when writing complex answer judging routines. When writing routines as complex as sentence open-answer judging, the code became complex and unwieldy, though not unusable. This type of judging could have been written much more efficiently in a language suited for intricate judging, such as Lisp. Although the version of Pascal chosen allowed other languages, like Fortran, to be combined with Pascal, Lisp could not be implemented at the time.

Another disadvantage in developing CAI lessons, it was found, was the lack of good CAI programming tools. Many tools exist that aid in programming, but not many are specific for CAI development. For this reason, many computer-based educational developers have written their own tools to do CAI functions. Few of these are available to other developers. More sharing of tools and utilities would benefit all concerned.

As mentioned before, OCBI wrote many utilities for use in development. In this way, OCBI was able to structure the programming staff to its greatest potential. Programmers who were just learning Pascal were given the task of creating displays with a graphics editor utility, shown in Figure 3. They could input all of the text and graphics needed for the lessons then combine all of the displays with another utility especially made for sequencing. Questions were added in a templated format that made it easy for the learners. The more experienced programmers were, in the mean time, writing the complex code needed for simulations and out-of-the-ordinary parts of the lesson. At all times, all programmers could rely on a team analyst to help with programming problems and queries. Each lesson also had a coordinator associated with it. The coordinator worked with the lesson

BEST COPY AVAILABLE

designer, the programmers, and OCBI to make sure everything ran smoothly. With this structure to aid in Pascal programming, the lessons were finished within their alloted time line. This type of staff structure is displayed in figure 4.



The Sedimentology of Flood Deposits

by Dr. James E. Pizzuto

Copyright © 1985 by the University of Delaware.
All rights reserved.

Figure 3. The graphics editor utility.

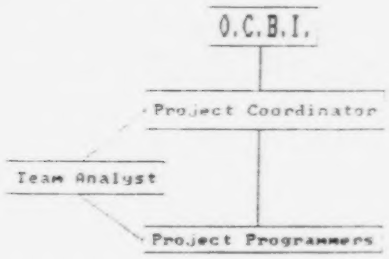


Figure 4. Staff structure of programming projects.

lternatives

There are some languages specifically targeted for CAI development on the market today. All vary greatly in their capabilities and compatibility. Authoring systems or authoring languages can be limiting. Often a user can only work within the bounds of the authoring language's capabilities. This can be disastrous if the lesson design calls for a feature not supported by the authoring language. Ideally, if an authoring language is being considered for lesson development, this language should allow the inclusion of code written in other languages. Sections of the lesson that call for capabilities beyond the bounds of the authoring language could be written in another language and incorporated into the lesson.

Authoring languages may be less portable than a standard language. Many authoring languages are written for one machine or only run under one operating system. Other authoring languages offer "scaled down" versions for different machines. This could result in a "scaled down" version of the original lesson or make development impossible for a second machine.

Finally, authoring languages can often be expensive. This is important to keep in mind if the budget is limited. Standard languages, including Pascal, are usually much less expensive and combined with all other features, a better buy in the long-run.

Conclusion

There are many advantages to choosing Pascal over other programming languages. It is easy to learn and use and it is usually well documented. It is a very well established programming language that is supported on several micro computers and many mainframe computers. Because many people are familiar with Pascal, there is an abundance of programmers and consultants. Each of these qualities is an advantage when developing CAI lessons for a microcomputer.

There are many languages and many microcomputers. When choosing a specific programming language for development, all the positive and negative qualities of the language must be weighed before a decision can be made. If the language is a well-rounded langauge, capable of accomplishing a specific CAI lesson without hindering the development, then it should satisfy all specific requirements for completion of the lesson.

BEST COPY AVAILABLE

TOOLS FOR DEVELOPING A GEOGRAPHY PACKAGE IN UCSD p-SYSTEM

Carol Jarom
Paige Vinall

Office of Computer-Based Instruction
University of Delaware

ABSTRACT

As part of the University of Delaware's commitment to CBE, the Department of Geography and the Office of Computer-Based Instruction developed "Map Layout and Design," a package of exercises on the IBM Personal Computer. The purpose of this package is to improve instruction in cartographic design and map layout through computer-aided design methods. The creation of a map layout is normally a very tedious undertaking involving hours of drafting maps. However, with the use of this set of exercises a student gains hands on experience in designing a map layout in a fraction of the time. A layout that would take ten hours by hand can be achieved in one to two hours with these exercises. This software package allows the student to create better maps in less time.

In addition to these exercises, the project has developed several useful authoring tools. These tools could be utilized to develop further CBE material in other disciplines.

INTRODUCTION

As part of the University of Delaware's commitment to CBE, the Department of Geography and the Office of Computer-Based Instruction developed "Map Layout and Design," a package of exercises on the IBM Personal Computer. In addition to the useful cartography exercises developed by this project, two very versatile lesson development tools have emerged. It was necessary to develop these tools in order to start creating the exercises, however as the exercises increased in complexity the tools were continuously changing and improving too.

The exercises progress from a simple layout involving only a map outline and page frame to a multi-element map layout. These exercises prepare the user to work with the Interactive Map and Graphics Editor (IMAGE) utility also provided with this package. IMAGE allows the user to create their own displays and edit the elements until the desired layout is achieved. Professors can tailor lessons to their own liking with this utility.

The following is a listing of the exercises.

1. Getting Acquainted
   This tutorial introduces the user to the IBM-PC keyboard and the general mechanics of the exercises.
2. Exercise One: The Page and Map
   The first exercise contains an introduction to the overall purpose of the package and a one element layout exercise.
3. Exercise Two: Adding a Title
   In this exercise users are presented with a more complex layout where they must concentrate on the balance between the title and the map placement within a given page frame. The user is provided with the ability to save layouts to disk.
4. Exercise Three: Titles and Legends
   This exercise increases the complexity of the layout design by adding two additional elements, a subtitle and a legend.

5. **Exercise Four: Introduction to Digitizing**
   Exercise four is a walk through exercise of creating a map and title using the digitizer tablet. Unlike previous exercises, the users enter their maps using a digitizer tablet rather than working with a prepared map.
6. **Exercise Five: Name Placement**
   In this exercise the user practices with the interactive placement of place names on states and countries.
7. **Exercise Six: Introduction to the Interactive Map and Graphics Editor**
   This tutorial explains and demonstrates the features of IMAGE provided with the package.
8. **Interactive Map and Graphics Editor (IMAGE)**
   This utility allows the users to create and edit their own layouts. It supports the Summagraphics Bit Pad and the Hewlett Packard plotter. Graphics displays used throughout these exercises were created with this utility.

DEVELOPING THE TOOLS FOR AUTHORING THE PACKAGE

Introduction to Developing the Tools

During the initial analysis of the project it was decided that certain tools would be needed to create graphics displays. Stand-alone utilities were developed to enter the map outlines and symbols from the digitizer tablet to data files, and to convert the data files to graphics on the screen and plotter. An additional utility was developed for creating and editing an entire layout on the screen. These utilities were combined to form IMAGE. Additional features such as zooming and saving layouts to disk were added as needed. Using IMAGE, the graphics displays for the exercises were created and saved.

The Interactive Map and Graphics Editor

IMAGE is an interactive tool that allows the user to enter maps, symbols and text objects; manipulate them on the screen; save the layout to disk or make a copy with the plotter. To enter maps and symbols the user traces the outline on the digitizer tablet. Text objects can be produced from a set of fonts by entering them on the keyboard. Within IMAGE each object in the layout is associated with a window. Several windows may be grouped together within a window. When the outer window is manipulated all of the windows contained in that window are also affected in the same manner.

Figure one is an example of the available fonts. Figure two illustrates nested windows.

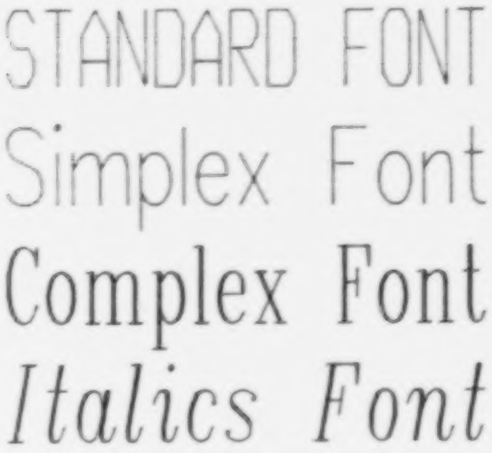


Figure 1. Fonts

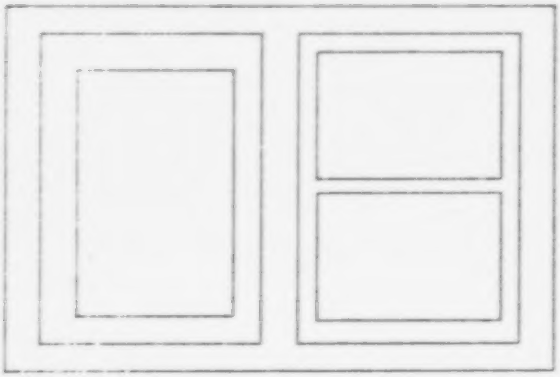


Figure 2. Nested windows

The key function of this software package is to allow users to manipulate the elements of a map layout in order to create a well balanced design. Therefore it was necessary to include many editing routines. These routines include the following:

* Move -- the ability to move an entire object
* Size -- the ability to increase or decrease the size of an object
* Rotate -- the ability to rotate the object on its axis
* Stretch -- the ability to leave one corner fixed while changing the object's size by selected corner.

These are the four basic editing tools used. However IMAGE provides many other helpful functions. These include features that allow the user to zoom in on one part of the layout, save and retrieve layouts, add and delete previously saved objects and so on.

Figure three shows a sample screen display from Exercise Five: Name Placement. To the right of the map layout is a list of the user's current options. To edit a state name the student uses the cursor keys to highlight the desired state abbreviation. Then the student may move, size, rotate, or stretch the state name. After the user is satisfied with the layout it is saved to disk and ready for the instructor to grade.

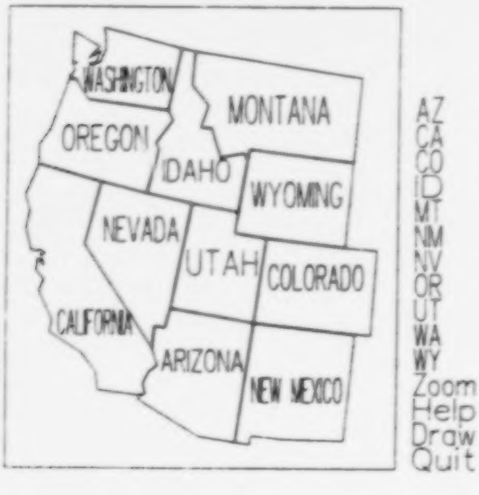


Figure 3. Sample layout

Text Writer

The next tool was developed to display text and graphics together in the exercises. This screen display creator is called Text Writer. Text Writer accepts as input a file name. The file contains the text to be displayed and embedded formatting commands. The formatting commands control text and background color, text positioning, paging and graphics layouts developed with IMAGE. A call to the subroutine text writer with a given file name produces screen displays within an exercise. Text Writer can be used to show a series of screen displays within one text file. These displays can be viewed without ever having to compile the exercise by executing a program that prompts the user for the file name of the displays to be viewed. A user could create a slide show type lesson without ever doing any programming or compiling. It's as easy to do as word processing. In this manner, an instructor could customize our exercises to their needs by changing the maps displayed and the accompanying text.

Figure four is an example of an input file to Text Writer. Note the embedded formatting commands. While Figure five shows how it appears on the screen.

Figure 4. Input file

BEST COPY AVAILABLE

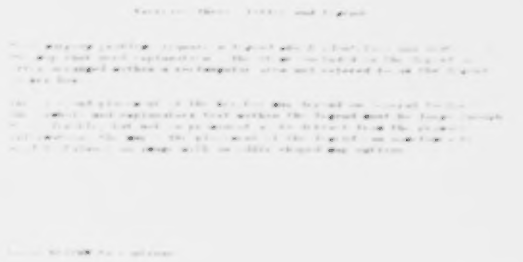


Figure 5. Screen display

Figures six and seven demonstrate how changing the video_mode command and the set_box command change the screen display to forty column mode. Figure six is the file with the embedded commands. Figure seven shows how it appears on the screen.

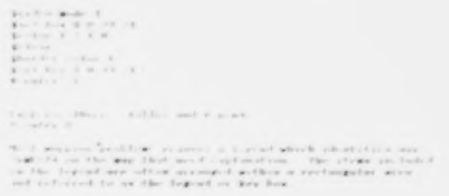
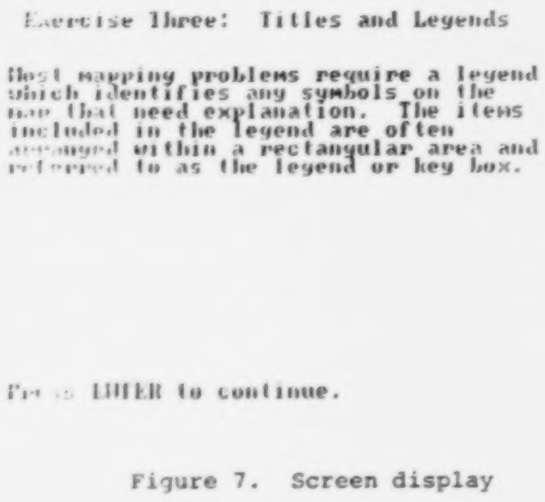


Figure 6. Input file



```
Exercise Three:  Titles and Legends

Most mapping problems require a legend
which identifies any symbols on the
map that need explanation.  The items
included in the legend are often
arranged within a rectangular area and
referred to as the legend or key box.




Press ENTER to continue.
```

Figure 7. Screen display

USING THE P-SYSTEM TO DEVELOP CBE MATERIAL

At the time the project was started, DOS and the p-System were the two major operating systems available for the IBM family of personal computers. The p-System was chosen for this project because of its user-friendly interface and portability. The user-friendly interface was particularly important since there is a high rate of turn over among our student programmers. The p-System is also helpful for professionals who use many different operating systems over the course of a week. The p-System displays all of the user's current options at the top of the screen and all of the options can be started with one key press. Although portability was not the main concern, different IBM and Apple projects at the Office of Computer-Based Instruction have exchanged useful subroutines. In the future this software package may be converted to the Apple Macintosh.

The p-System also provided an organized way to divide programs up into separately compiled units. Using units the project was divided between several programmers and each unit was developed and tested independently. These units were then placed together in a library to create an initial version of IMAGE. The programmers were in charge of making all necessary changes to their units so that different versions of the same code would not exist.

Another important feature of the p-System is the dynamic and automatic management of code segments. If a program is too large to fit into memory units are swapped into the memory as they are needed.

The p-System supports Pascal, FORTRAN, and Assembly. This software package was written mostly in Pascal with a few Assembly routines. Pascal lends itself to good modular programming. Most of the student programmers at the University of Delaware already have a strong background in Pascal. The strict type checking of the language makes the code easy to read.

The p-System was originally developed to support Pascal. Due to the design goal, the p-System compiles Pascal programs very rapidly into a pseudo machine code called p-code. At run time, the p-code is interpreted by the

BEST COPY AVAILABLE

p-System machine emulator. The p-code is transportable to different computers which have their own p-System machine emulator.

Overall the p-System has provided a strong foundation for the project, but some disadvantages were found. The primary disadvantage is a slow execution speed caused in part by time needed to interpret the p-code at run time. Separately compiled units are linked together at run time which also causes an initial delay when programs are executed.

Several steps were taken to increase the speed of the programs. First, the code was streamlined to remove redundant and loop invariant code. Second, recursive routines were replaced with iterative routines wherever possible. Iterative routines save the computer from having to pass parameters on the stack. Third, runtime error checking was performed explicitly by our software rather than by the p-System. Default range checking is performed at every access to an array. If the software is properly debugged these checks are unnecessary. The same is true with input and output error checking. Finally, routines that would often be used at the same time were grouped together in the same unit and underlying routines which were used by all of the routines were grouped together. These simple improvements increased the execution performance by about seventy-five percent. With the p-System's support of the IBM 8087 math co-processor, the software runs at an acceptable speed.

Another disadvantage of the p-System is the lack of high speed graphic routines. The highest level graphic routine provided draws a line between two points. To deal with this problem, the project programmed the needed routines and simplified some of the lesson design requirements.

CONCLUSION

This project has developed useful tools that could be used not only for this geography project. IMAGE and Text Writer provide the necessary authoring tools to developed CBE in any discipline. The p-System provided a user-friendly environment for the development of CBE material, and peripheral devices made a major contribution to this educational package.

COMPUTER USER GROUPS: AN UNTAPPED RESOURCE

LaVerne L. Ludden, Ed.D.
Assistant Professor
Department of Supervision
Indiana University - Purdue
University at Fort Wayne
2101 East Coliseum Boulevard
Fort Wayne, Indiana 46805

ABSTRACT

The purpose of this study was to explore and describe the organizational structure of computer user groups and what roles they play in adult computer learning. The research consisted of two stages that used triangulation methodology.

The pilot study involved a survey of IBM Personal Computer users in northeastern Indiana, which used a 34 item questionnaire, answered by 128 persons. The purpose of this initial study was to examine what degree of importance members of a local computer user group attached to the learning function of the group and to yield data about the characteristics of group members. The second stage consisted of a national survey of computer user groups, which used a 57 item questionnaire, telephone interviews, and content analysis of newsletters. The nationwide survey was answered by 72 groups, representing 55,222 members.

Five major findings resulted from the study. These included:

1. Computer user groups were a major provider of computer learning for adults in the United States with the emphasis placed on practical applications of the learning.
2. Computer user groups were one form of autonomous learning groups.
3. Computer user groups existed in all parts of the United States and tended to have a formal organization.
4. The typical member of a computer user group is a male, aged 35-40, is a college graduate, has an average income between $30-40,000, was employed in a professional job, and had between 2-5 years of experience in the use of computers.
5. Members of computer user groups preferred self-directed learning and peer led learning to classes at educational institutions.

In addition to satisfying the learning needs of members, the groups provided supplemental services that supported other computer related needs of the members. The results of this study suggested that the role of such groups in adult learning may be far more important than was generally recognized by educators.

COMPUTER USER GROUPS: AN UNTAPPED RESOURCE

INTRODUCTION

Computers are becoming an integral part of our society. They are used in a broad spectrum of applications from recreation to work. It is inevitable that within the near future every individual in the United States will use computers to perform common daily tasks. Computers will change our work (Pask & Curan, 1982), our leisure activities (Frude, 1983), and even the way that we think (Papert, 1980).

Educational institutions have been engaged in the development of new strategies, educational technologies, and curriculum to meet this challenge. Their basic thrust has been aimed at students in the elementary, secondary, and post-secondary levels. This approach overlooks the fact that a majority of the population needing basic training and education in computers is adults not in school. Business and education have begun to provide education to adults but an even greater effort needs to be made in this area.

This paper examines a resource that has been used by hundreds of thousands of adults to learn about computers. This resource is the computer user group. Computer user groups have over 350,000 members in the United States and exist in small communities as well as metropolitan areas (Makower, 1984). The research summarized in this paper was the culmination of the dissertation work of the author. The paper examines the role of personal computer user groups in adult continuing education. Professional educators can use this information to support and establish computer user groups to assist individuals in becoming computer literate. The results of this research should help educators recognize the significance of education that takes place outside the formal school setting and identify methods which can be used to build on this resource in an effort to improve computer learning in our society.

REVIEW OF LITERATURE

The history of the United States is replete with examples of independent or autonomous groups that have formed to advance the education of adults; among these groups are the Philadelphia Junto, Lyceums, Chautauqua, and Great Books programs. An autonomous learning group has education as its primary or facilitative role, where the group meets frequently, and the group accepts the responsibility for planning learning activities. The average adult can quickly identify numerous groups that meet this definition. Stamp clubs, ham radio clubs, literary clubs, Bible study groups, and computer user groups are all examples of autonomous learning groups.

Over 75% of all adults carry out independent or self-directed learning projects every year (Penland, 1979). The phenomenon of self-directed learning has been well studied by adult educators. However, very little attention has been focused on self-directed or autonomous learning groups. Allen Tough (1979) has noted that "As we become more and more aware of the frequency and importance of autonomous groups, perhaps we will develop more effective ways of helping them plan and conduct their learning." (p. 146).

There was only one reference to computer user groups identified in professional journals, magazines, and dissertations abstracts. However, a few books and several popular magazines and newspapers contained a great deal of information about computer user groups.

Naisbitt (1982) has noted three trends in our society that relate to computer user groups. He states that we are moving from high tech to high touch, from institutional help to self-help and from organizing through hierarchy to organizing through networks. Turkle (1984) has indicated that computer user groups sprang up around the country and were "imbued with excitement not only about the computers themselves but about the new kinds of social relationships people believed would follow in their wake." (p. 171). This reinforces what Naisbitt had to say about the move toward high touch. The literature suggests just how much computer user groups are representative of these changes in our society.

Computer user groups are typically organized to support computer users of a specific type of personal computer or software application (Shea, 1983). They appear to serve the learning and problem solving needs of their members with regard to computers (Metzger, 1983). According to Radding (1983) the problem solving takes place in a supportive atmosphere where other members have experienced similar problems and are willing to provide help. The groups ranged in size from a few friends that meet to discuss computers (Shea, 1983) to the Boston Computer Society with 15,000 members.

The education and problem solving activities of the group are supplemented by other member services. An important service is the newsletters published by most computer user groups (Makower, 1984). Most groups also maintain libraries of computer software that is in the public domain and distribute this software to members. The groups provide members with useful consumer advice (Froelich, 1984) and serve as cooperative buying clubs to obtain lower prices on computers and supplies for members (Mace, 1984).

The Washington, D.C. IBM-PC computer user group surveyed its membership and found that they were typically male, college educated, approximately 40 years old, and have a family income of about $45,000 (Foxwell, 1984). These results were similar to the survey of members who replied to a newsletter survey for the Boston Computer Society (Rotenburg, 1984). Most of the groups have monthly meetings (Froelich, 1984). A predominate feature of most of these meetings is a question and answer period in which members freely ask questions and receive answers from another member of the group. The groups are sometimes associated with businesses or schools. Christ (1985) describes a computer user group that is supported by the learning assistance center of a public university.

The literature review revealed a great deal of information about computer user groups. However, all of the information basically came from journalistic reports and were not based on scholarly research. Therefore, this study was undertaken to obtain more information about the groups and discover information that would be of value to educators.

RESEARCH DESIGN AND METHODOLOGY

The research involved a multistage approach that used triangulation methodology. A pilot study was conducted in 1984, which involved a survey of the members of the Northeast Indiana IBM-PC Club. The purpose of the pilot study was to determine if a computer user group was a primary source of computer learning for its members. A questionnaire was developed which would determine the demographic characteristics of members, identify the sources they used to learn about computers, and determine their reasons for joining the group. Questionnaires were distributed to 323 persons on the group's mailing list, which contained the names of both members and non-members. Approximately 150 persons were members, and of this group, 83 returned the questionnaires for a 55% response rate.

The pilot study demonstrated the importance of computer user groups in providing their members with education about computers. Therefore, a nationwide study of computer user groups was undertaken. The purpose of the nationwide study was to obtain information about the organizational structures of computer user groups and what roles they play in adult computer learning. The study included a survey of computer user groups, telephone interviews with officers of computer user groups, and content analysis of newsletters of various groups.

The survey included the distribution of 318 questionnaires to computer user groups. A sampling frame was impossible to identify since no one organization keeps a record of all computer user groups. Therefore, a judgmental sample rather than a random sample was chosen that was based on geographic location and type of computer used by the group. There were 72 or 23% of the questionnaires returned and considered valid for data analysis. However, the groups represented were from 31 (62%) of the 50 states. Also, the groups responding represented users of 26 different computers made by 14 separate companies which accounted for 96% of the most popular personal computer models (Honan, 1985). The rate of return is less important in a judgmental sample and the fact that the basic requirements of the judgemental sample were satisfied indicates that the data is valid although it cannot be assumed to be representative of all computer user groups in the United States.

Telephone interviews were conducted with 10 officers who had responded to the survey. The interviews lasted from 20 to 40 minutes. The interviews were conducted with people from all regions of the United States and individuals who represented the various organizational types of computer user groups.

A total of 71 newsletters were subjected to content analysis. The newsletters came from groups all over the country including groups who responded to the survey and included a year's supply of newsletters from five of the groups who had responded to the survey.

FINDINGS

There are five major findings that have been identified from this study.

These findings include:

1. Computer user groups were a major provider of computer learning for adults in the United States.

An estimate of the number of computer user group members was cited in the literature review as being at least 350,000 members and perhaps "many times that amount" (Makower, 1984). The national survey seemed to confirm this estimate and indicated this estimate might be quite conservative. The 71 groups which responded to this survey had over 55,000 members and these groups represent less than 10% of all identified computer user groups. Therefore, an extrapolation of this data indicates the possibility that there are nearly 600,000 computer user group members in the United States.

The national and local surveys, telephone interviews, and content analysis of newsletters all confirmed that the primary purpose of computer user groups was to assist their members in learning more about computers. The groups typically held monthly meetings, conducted classes, and published newsletters in carrying out their educational purpose.

The data indicated that not only did computer user groups exist for the purpose of helping members learn more about computers but members joined computer user groups primarily for the objective of learning more about computers. Their motivation for the learning appeared to be connected more with their own curiosity and interest rather than an outside stimulator, specifically an employer.

The learning that took place was practically oriented and was typified by the term "problem solving". Members engaged in a great deal of informal interaction outside of the groups' formal meeting and activities to learn about computers. The group meetings typically involved a question and answer period where members alternated between the roles of learner and teacher. The objective of this activity was to find solutions to problems that members encountered in using their computer, peripheral equipment, and software.

Members of computer user groups appeared to be satisfied with the learning that occurred within the computer user groups. The meeting, which was the basic method used by computer user groups to assist their members in their learning efforts, was typically held once a month for about 2 1/2 hours and was led by the members of the group.

2. Computer user groups were one form of autonomous learning groups.

Autonomous learning groups were defined as: a group which has education as its primary or facilitative role, where the group accepts the responsibility for planning learning activities. Computer user groups met the criteria established by this definition. The previous finding stated that the primary role of computer user groups was educational. They were composed of members who volunteered their time to organize the group, maintain its activities, and plan learning activities.

The participation of members differed considerably from one group to the next. However, it appeared that it was necessary for the computer user group to have a "core group" of members who accepted the responsibility for promoting, administering, and carrying out the tasks of the entire group. This core group was the catalyst that kept the group functioning but the core group was constantly changing. In other words, the members who were a part of the core group changed with old members being replaced by new ones.

3. Computer user groups existed in all parts of the United States and tended to have a formal organization.

Computer user groups were identified in each of the fifty states, indicating that they were not a localized phenomenon but were quite widespread. They averaged approximately 100 members but were as small as 6 members and as large as 28,000 members. Most groups served computer users of one type or family of computers like Apple or IBM but a few were generic.

The groups typically had officers, collected dues, expended funds to further their purpose, were sometimes incorporated, held regular meetings, and engaged in many activities and services that ranged beyond what would typically be considered educational.

4. Members of computer user groups were typically over 35, had a high family income, were college graduates, and were male.

While this was the typical profile of the computer user group member, it is interesting to note that the study found a wide range of member characteristics. The groups were not the exclusive domain of adults. In fact, some groups listed members who were only 7 years old. On the other hand, members participated who were as old as 88 years of age.

The higher incomes of members may not be unusual considering that people need a higher income to afford a computer, which was basically a luxury item. However, computer user groups had members who did not own computers but used them in school or at work. Also, previously established research has demonstrated that a correlation existed between income and education, which might explain why user group members had a higher level of education. While male members outnumbered female ones almost 9 to 1, it appeared that there was a growing trend toward increased female membership in the groups.

5. Members of computer user groups preferred self-directed learning and peer led learning to classes at educational institutions.

Almost all of the computer user group members indicated that they had used self-education as a method for learning about computers. Nearly half of the members relied on the computer user group as a source of learning. Members further indicated a strong preference for self-directed learning as their primary source of learning in the future. However, the number of members who looked at the user group as their primary source of learning in the future was the same as those who expected classes at colleges to be their primary source. There were no other educational institutions that members planned to use as a primary source of learning in the future.

The leaders of computer user groups perceived their group as satisfying most of the computer learning needs of their members. Further, they felt that members preferred to learn about the computer through the user group rather than a class offered by an educational institution.

IMPLICATIONS

Autonomous learning groups have been recognized by few, but influential adult educators such as Brookfield, Houle, and Tough, as important manifestations of adult education in our society. Autonomous learning groups and self-help groups have a great deal in common. In fact, it would appear reasonable to consider autonomous learning groups to be a type of self-help group whose purpose is to meet the learning needs of individual members.

Computer user groups are a type of autonomous learning group. They are community based groups functioning outside the domain of educational institutions. Computer user groups are the focus for a significant amount of computer learning that occurs within a community. Therefore, this study on computer user groups has several implications for the adult and community educator. The most important of these implications are explained in this section.

1. The existence of computer user groups demonstrates the scope and importance of autonomous learning groups.

This study has found that hundreds and perhaps thousands of computer user groups are organized throughout the United States. There are hundreds of thousands of people who belong to these groups. The groups have collected and spent millions of dollars annually on the learning activities of members. Computer user groups have been formed by individuals attempting to adapt to the introduction of new technology in society.

Computer user groups were not the only groups that have been formed to assist people to adapt to new technology. Paioff (1984) described the formation of the American Automobile Association (AAA), which originated out of a merger of local automobile user groups. These local automobile clubs were started with the purposes of educating members and the community about the automobile and advocating better laws for the users of automobiles. Rocket clubs and ham radio clubs are other examples of the high technology learning group. Robot user groups are currently in the process of being organized around the country.

2. Adult and community educators can establish a mutually beneficial association with autonomous learning groups and computer groups in specific.

Adult and community educators can identify autonomous learning groups in their communities and begin to work with them. Such groups represent an identifiable number of learners who could be recruited for classes offered by educational institutions. Further, the members of computer user groups can provide meaningful data about the needs of learners related to computers. By including members of computer user groups in a needs assessment, it is quite possible that an educational institution would be able to develop classes that appeal to a large number of computer users.

The members of computer user groups also could be recruited to teach classes for continuing studies programs or regular classes. These individuals possess a great deal of expertise in microcomputers. Recruiting instructors from computer user groups would also have the benefit of the person being able to recruit students for classes from among other members of computer user groups.

Computer user groups could share equipment with educational institutions. The groups could also provide a public domain software library that could be of use to educational institutions and their students. Computer user groups could assist the educational institution in expanding its services to students by maintaining a computerized bulletin board system, coordinating computer fairs, and perhaps providing experts who would be willing to tutor students or answer some of their questions.

Educational institutions could provide a valuable service to the community by reaching out to computer user groups. Providing assistance to computer user groups would be a highly effective public relations activity. Educational institutions could provide a facility for meetings, maintain a group's public domain library, assist in the publication of the newsletter, refer people to the group, allow the group to use computer equipment and assist the group in planning its learning activities.

3. Computer user groups provide some basic guidelines that adult and community educators can follow in pursuing improvements in their educational programs.

There were a number of features about computer user groups that appeared to contribute to their success and which could be adapted by adult and community educators. Many of the ideas presented are already used by adult and community educators but other ideas may be new to many practitioners. It is important to approach each concept with the understanding that they may not fit into the current structures of adult and community education programs. In fact, it may suggest ways that entire programs of adult and community education can be restructured. These features include:

a. Computer user groups relied on the expertise of the group rather than one individual, such as a teacher.

A group's effectiveness comes from the group's ability to respond to the learning needs of all members. This is accomplished because the group itself is expected to respond to an individual's needs. Few educators, even those who embrace androgogical methods, are willing to establish a system of learning where the students are the major source of expertise.

b. There was a problem solving orientation to the learning that took place in computer user groups or what can be perceived as a high degree of relevance.

Members of computer user groups turned to the group for solutions to problems that they experienced in using their computers. The newsletters were written to help members learn to use their computers more productively. Adult educators are aware of the need for relevancy in adult learning. This study reinforces the importance of this method.

c. Much of the learning in computer user groups took place during informal interaction and was based on the social needs of the members.

Adult educators could make their classes and workshops more attractive to adults if they provided more opportunities for the students to develop a feeling of camaraderie and learn from personal experiences with each other. Seminar and workshop instructors who have observed what takes place during coffee breaks or evening social activities, can appreciate this feature of computer user groups. It is possible to plan for social activities and use them effectively to increase the learning that takes place in an educational program.

d. Computer user groups provided ongoing support for their members that did not end when a quarter or semester ended.

A keyword in the vocabulary of adult and community educators is lifelong learning. However, this is often interpreted to mean that adults will be encouraged to participate in educational activities their entire lives and we as educators will provide classes, workshops, or seminars that they can conveniently enroll in whenever they want. One lesson taught by computer user groups is that people want continuous support in their efforts to learn particularly when the learning involves a new technology. Members may not always take advantage of this learning but it is important to know that the assistance is available whenever it is necessary.

e. Members were allowed to participate at their convenience and according to their needs.

Members of computer user groups were not required to attend meetings of the group or of special interest groups. They did not have to volunteer to provide services or assist in group activities. Considering the many demands that jobs, family, and social interests place on the adult, it only seems reasonable that this feature would be attractive to adults.

f. A wide range of services were provided to members that ranged beyond what could be considered learning activities but they were important to the members of the group.

Computer user groups did not narrowly define their role as an agent of learning and ignore the other needs of their members. By being attentive to needs that existed beyond the learning needs of adults they were able to attract and keep more members. They did this with services and activities that complemented the learning needs and were not entirely divergent from the basic purpose of the group. However, this approach suggests that adult and community educators might be able to develop programs with a broader and more intense appeal to adults if they would supplement educational services with related services that meet other needs of their students.

g. A small fee was charged for the services of the group but a great deal of reliance was placed on members to volunteer their service to offset normal staff costs associated with the operation of a program.

The user groups were able to charge small membership fees because the groups were led and operated by volunteers. Some groups reduced the cost of membership fees for those members who volunteered their time. It would be possible for adult and community education programs to operate in the same fashion, although the basic administrative fees for running the entire operation would probably require a higher membership fee.

h. Credit for learning was not always important for adults but rather they were motivated by the need to learn and solve problems.

The members of computer user groups joined to learn but only expected the group to provide them with knowledge and answers to problems. There was not an expectation of credit that could be used to enhance a resume or provide evidence to an employer of continued learning. The bottom line was improved performance and skills. Adult and community education programs can be more innovative in their programming if they are not always bound by an educational institution's requirements for awarding credit.

4. Computer user groups provide adult and community educators with a model that can be used to help the general populace become more computer literate.

There is a great deal of debate taking place about the definition of computer literacy and the need for people to become computer literate. However, assuming that an adult or community educator agrees with the premise that computer literacy, whatever it might be, is needed by people, then computer user groups provide a model for promoting computer literacy.

The computer user group can be supported with a minimal amount of effort and resources from an educational institution. The members of the group provide the expertise, leadership, and volunteer time needed to carry out the goals and objectives of the program. The group can accommodate individuals with a wide range of learning needs.

5. Computer user groups provide a model that can be used by adult and community educators to help society adapt to future introductions of state of the art technology.

The rapid development of new technology in our society means that people can expect to learn and adapt to technological innovations throughout their lives. Computer user groups offer a model that can be used by adult and community educators to aid individuals in learning to use new technologies.

Informal groups can be organized by educational institutions to assist users of the new technology to learn about the technology and solve problems that they encounter in its use. By taking the lead in the organization of these technological user groups, educational institutions will provide a valuable educational service to the community and can perhaps speed the rate of acceptance and utilization of new technology.

SUMMARY AND CONCLUSIONS

Computer user groups were organized primarily to assist their members in learning more about computers. In addition to satisfying the learning needs of members, the groups provided supplemental services that supported other computer related needs of the members. Computer user groups were one form of autonomous learning groups and the results of this study suggested that the role of such groups in adult learning may be far more important than was generally recognized by adult and community educators. Adult and community educators should also consider the features of computer user groups that could be incorporated into their programs to improve those programs.

The continuous and rapid development of new technology will place increasing demands on individuals to change and adapt to this new technology. More importantly, individuals should learn to control the technology rather than react to it. Computer user groups provide a model that can be duplicated by adult and community educators in helping people to take control of the technology and use it to improve their lives.

REFERENCE LIST

Christ, Frank L. (1985). Techtalk: improving knowledge of computers on your campus through user groups. Journal of Developmental Education, 8(1), 22-23.

Foxwell, Harry J. (1984, July 10). A capital survey. PC Magazine, pp. 359-363.

Froehlich, Robert A. (1984). The free software catalog and directory. New York: Crown Publishers.

Frude, Neil (1983). The intimate machine: close encounters with computers and robots. New York: New American Library.

Mace, Scott (1984, September 17). Users groups reach out. InfoWorld, pp. 20-24.

Makower, Joel (1984, October 9). A growing network for owners: computer clubs offer access and help. USA Today, p. 3D.

Metzger, Elizabeth (1983, March). The helping hands of computer clubs. Personal Computing, pp. 142-147.

Naisbitt, John (1982). Megatrends: ten new directions transforming our lives. New York: Warner Books.

Papert, Seymour (1980). Mindstorms: children, computers, and powerful ideas. New York: Basic Books, Inc., Publishers.

Pask, Gordon & Curran, Susan (1982). Microman: computers and the evolution of consciousness. New York: MacMillan Publishing Co., Inc.

Penland, Patrick (1979). Self-initiated learning. Adult Education, 29, 171-179.

Radding, Alan (1983, April). Computer clubs to the rescue. Popular Computing, pp. 152-153.

Rotenberg, Jonathan (1984, January/February). The view from center plaza. Computer Update, pp. 6-7.

Shea, Tom (1983, June 13). Get a micro and then find a users' group. InfoWorld, p. 24.

Tough, Allen (1979). The adult's learning projects: a fresh approach to theory and practice in adult learning. San Diego: Learning Concepts.

Turkle, Sherry (1984). The second self: computers and the human spirit. New York: Simon and Schuster.

DEVELOPING A LIBRARY SKILLS PROGRAM FOR THE IBM PC

Deborah E. Richards

Office of Computer-Based Instruction
University of Delaware

ABSTRACT

A package of four PLATO lessons was converted to run on the IBM Personal Computer with a monochrome display. To convert the lesson, a screen editor was developed that would allow non-programmers to create individual displays of a lesson, containing text and character graphics, which would then be translated into Pascal code by the editor. After the individual displays were completed, sequencing and answer judging routines were added to create a final working version of the lesson. This paper reviews the conversion process and discusses the advantages of the approach used.

INTRODUCTION

In 1981, the University of Delaware Library chose computer-assisted instruction (CAI) to replace a more traditional approach to teaching basic library skills to freshman English students. At that time, reference librarians were giving approximately 150 course-integrated bibliographic lectures to freshman English classes each year. CAI was a logical and convenient choice to replace these bibliographic lectures because of its obvious instructional advantages and because CAI was already well-established and supported at the University of Delaware.

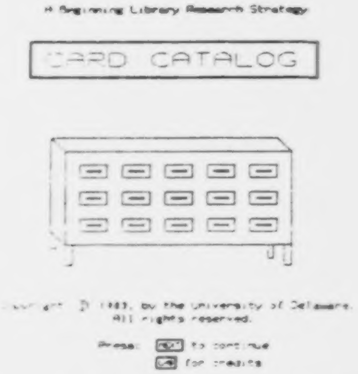
Librarians at the University of Delaware Library, working with CAI professionals from the Office of Computer-Based Instruction, developed five tutorial lessons for the PLATO network. "The Card Catalog" explains how the card catalog is used to locate books by author, title, and subject. The use of various periodical indexes to find articles on specific topics is taught in "Periodical Indexes." "Newspaper Indexes" explains how to find newspaper articles on specific topics. "Government Documents Indexes" explains how to locate information published by the U.S. government by using government documents indexes. An additional lesson gives information on the physical location of books, periodicals, newspapers, and government publications in the University of Delaware Library.

These lessons aroused a great deal of interest at the local, state, and national levels. From the many inquiries and requests for demonstrations of the lessons, it became obvious that many other libraries were interested in using CAI lessons, but lacked the time, resources, and programming expertise to develop the lessons themselves. Other academic libraries would have been interested in using the University of Delaware lessons except that they were specific to the University of Delaware Library and they were only accessible to libraries which had access to the PLATO network. In 1983, the decision was made to convert the library package to run on a microcomputer. In order to make the lessons available to a wider audience, they were redesigned to be more generally applicable by removing information specific to the University of Delaware Library.

THE CONVERSION PROCESS

The first step in the conversion process was to choose a target microcomputer. After much discussion, the IBM Personal Computer was chosen for several reasons. The IBM PC is a powerful and versatile machine which is increasing in popularity in college and university environments. Additionally, its use appears to be on the rise in college and university libraries. One factor which may have helped spark this increase in use is the decision by the Online Computer Library Center (OCLC) to use a modified IBM PC as its new Model 300 terminal. OCLC is a not-for-profit computer library service and research organization. It operates an international network that libraries use to acquire and process library materials, in addition to other services provided. The M300 terminal consists of an IBM PC with two disk drives and a monochrome display. Because the M300 is configured with a monochrome display and because most library applications are text-oriented, we decided to convert the lessons so that they could be used on the monochrome (text only) display.

The next step in the conversion process was to redesign the lessons. This included removing content specific to the University of Delaware Library, modifying the display layouts to fit on the 80 character by 25 line IBM monochrome display, and modifying the graphics used in the lessons. The number of characters per screen is roughly the same for the IBM monochrome display and a Control Data IST1 or IST2. Therefore, one display in the PLATO lessons usually converted to one display in the IBM PC version. Rather than eliminating graphics from the lessons, the IBM special characters needed to be used to produce graphics on the monochrome monitor. This meant that simple graphics could be drawn by aligning consecutive characters. For example, a box could be drawn by aligning a number of horizontal and vertical single characters and inserting corner characters. This limited graphics capability allowed us to create surprisingly good graphics and to retain many of the graphics used in the original lessons. Figure 1 shows a display with graphics from the original PLATO lesson on using the card catalog. In comparison, figure 2 shows this same display, modified for the IBM PC version of the lesson.

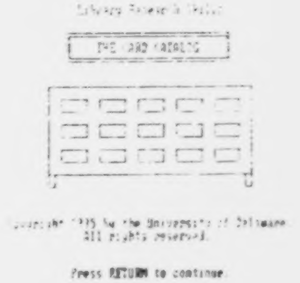

A Beginning Library Research Strategy

CARD CATALOG

Copyright © 1983, by the University of Delaware.
All rights reserved.

Press: [illegible] to continue
[illegible] for credits

Figure 1. Title display from the PLATO lesson.


Library Research Skills

THE CARD CATALOG

Copyright 1985 by the University of Delaware.
All rights reserved.

Press RETURN to continue.

Figure 2. Title display from the IBM PC lesson.

Some of the special characters and text features of the monochrome display allowed us to incorporate special techniques which were not possible in the original versions of the lessons. Three types of text could be used for emphasis: underlined text, bright text, and inverse text. The inverse text is especially effective on the monochrome display. This allowed us to improve several multiple choice quizzes in the original lessons. Figure 3 shows a multiple choice question from the original PLATO lesson where a student has been asked to identify a part of a reference. Figure 4 shows the same question in the IBM PC version of this lesson. In the new format, the student is asked to identify a part of the reference by using the cursor keys to move a highlighted bar to the correct item. This format was much clearer and more effective than the original multiple choice format because it had been difficult to write effective multiple choice questions requiring identification of specific elements in an example.

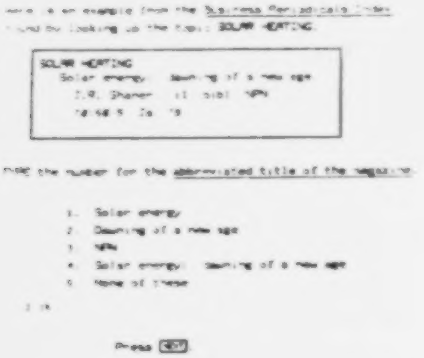


Figure 3. Multiple choice question from the PLATO lesson.

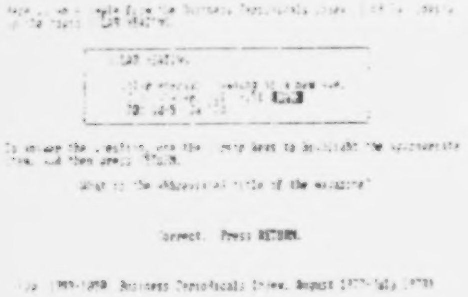


Figure 4. Multiple choice question from the IBM PC lesson illustrating the new format.

The inverse text was also effectively used to point out items that were being explained. Again, in some cases, this feature worked more effectively than the techniques used in the original lessons.

Once the redesign of the lessons was complete, the reprogramming could begin. The decision was made to program the lessons in Pascal. This language was chosen for several reasons. It is a structured language suited to CAI, it is flexible, and there is a readily available supply of skilled programmers. Because the lessons are tutorials, there is a large amount of text both in the body of the lesson and in the examples from the resources being explained. For this reason, and because all graphics had to be created using the special IBM characters, it would have been time consuming and inefficient to program, compile, and check each individual display. Therefore, the first step in the programming process was the development of a screen editor.

This editor enabled non-programmers to place text and graphics on the screen which would then be translated by the editor into Pascal code. Without any programming expertise, an individual could create character graphics, enter and position text, add, delete or reposition any text or graphics on the screen.

This project was the first conversion or development project undertaken at the Office of Computer-Based Instruction to utilize a screen editor. We served as a testing ground for the development of screen editors for both the monochrome and graphics monitors. Problems encountered during this conversion project helped shape features added to the editor. For example, in the earliest version of the editor, both text and graphics could be moved once they were created. However, one of the problems we encountered was that complex objects had to be moved piece by piece. This meant that each box and each line drawn had to be moved individually to shift an entire graphic on the screen. The solution to this problem was to allow the definition of compound objects that could be moved as a whole. Other features added to the editor as a result of this project were text editing features, the ability to use any of the IBM special characters to draw lines or fill boxes, and scanning for intersections of lines or corners of boxes so that corner and intersection characters could be automatically inserted. In addition to enhancements of the monochrome version of the screen editor, a color/graphics version was developed expanding and improving upon the original monochrome version.

Once all of the screen displays had been created for a lesson, the process of sequencing the displays and adding answer judging routines began. An experienced Pascal programmer wrote and debugged all of the more complex routines needed. Then, these routines were added to the displays for each individual lesson. Sequencing routines allowed movement forward and backward, one display at a time in the lesson. A feature was also added to allow users to return to a menu display from any point in the lesson. Answer judging routines included multiple choice answer judging and cursor choice answer judging.

BEST COPY AVAILABLE

ADVANTAGES OF THIS CONVERSION PROCESS

One of the main advantages of the process was the amount of programming time saved. The lessons had a large amount of text, both in the explanations presented and examples used. The shortest lessons consisted of approximately 35 individual displays while the longest lesson consisted of 85 displays. Because we had the screen editor, individual displays could be created by inexperienced programmers, checked by the lesson designer, and revised all without time consuming compiling and recompiling.

Another advantage of this process is that complex routines, such as answer judging and sequencing, were programmed and debugged by experienced programmers. Then, they were added to the code for the individual screen displays by less experienced programmers. This allowed us to have one experienced programmer who could serve as a consultant to the less experienced programmers on the development team.

A final advantage of this process was that debugging was easy and painless. Problems on individual displays could be changed before sequencing and answer judging routines had been added. Because the lessons were programmed in a structured language, errors are easy to find and the code is easy to maintain. In addition, it would be relatively easy to customize the lessons to another library setting.

CONCLUSION

In order to make library skills lessons developed at the University of Delaware available to a wider audience, the package was converted to run on the IBM PC. The process used to convert this package was to create individual screen displays using a screen editor, and then to link these displays together by incorporating sequencing and answer judging routines. This project was the first at the Office of Computer-Based Instruction to follow this procedure. The result was that development, revision, and debugging was much easier and took much less time than anticipated. This proved to be a successful and effective approach and has been used as a model for additional CAI conversion and development projects for microcomputers undertaken at the University of Delaware.

This project also proved that effective CAI, incorporating both text and graphics, could be presented using the monochrome display on the IBM PC.

SOME CONSIDERATIONS INVOLVED IN THE INSTALLATION OF ETHERNET NETWORKS AT THE UNIVERSITY OF DELAWARE

George A. Reed
Nancy Balogh

Office of Computer-Based Instruction
University of Delaware

ABSTRACT

The availability of microcomputer networks creates new opportunities for innovative uses of computers in educational settings. Considerations of network costs, installation requirements, capabilities, and limitations are keys in choosing a network application, but some of these questions are more easily addressed than others. This paper will discuss the implementation of Ethernet networks by the Office of Computer-Based Instruction at the University of Delaware.

INTRODUCTION

In 1984 the University of Delaware's Office of Computer-Based Instruction (OCBI), in conjunction with the College of Engineering, began the installation of two networks of IBM Personal Computers (PCs) in the departments of Mechanical and Aerospace Engineering and Chemical Engineering. These two sites were designed primarily for student use. An additional network, designed principally for the development of computer-based instructional materials, was installed in OCBI offices at the same time. A limited number of IBM PCs were already being used in each location, and the need for communications and resource-sharing capabilities was becoming increasingly apparent. It was anticipated that both educational and physical advantages could be derived from a network installation.

This paper discusses the attributes sought for our networks and the degree to which our installations meet these requirements. Considerations of cost, hardware and software needs, and installation requirements are detailed, and possible future developments are projected.

OBJECTIVES FOR THE NETWORK

Considerations for the two proposed engineering network sites included a combination of educational, physical and cost requirements. The following characteristics are a minimum for our requirements:

* The system must be transparent to users.
* Installation must not require special skills.
* System additions or reconfigurations must be easily implemented.
* A file server must be supported.
* Printers and other peripherals must be accessible from any computer on the network.
* Individual sign-ons must be possible.
* Security of files must allow different levels of access.

In addition, in order to meet our educational objectives, the following characteristics are desired:

* A mail facility should be available.
* Faculty should be able to leave assignments for students.
* Faculty need to have access from their offices.
* Students should be able to submit assignments.
* Complete and readable documentation is needed.
* A wide variety of software should be supported.

- A means of connecting both microcomputer networks into the College ofEngineering's existing network of minicomputers and mainframes is needed.
- A way of determining hours and types of use is desirable.
- User groups should be supported.

At the time our networks were installed, Ethernet could be implemented to satisfy the minimum requirements and all but the last three desired goals. The acquisition of software and hardware for connecting these networks into Engineering's Ethernet is underway at present, and network applications of software are being investigated as they become available. The goal of determining network usage depends on the further development of software, but that should become available within the near future.

COUNTING THE COSTS

The initial cost of networking microcomputers is substantial, but there are many compensating cost savings. Each installation will achieve varying benefits, depending on the use made of the network. A substantial saving has been obtained for our sites by the sharing of resources, including hard disk space and printers. An even greater savings is obtained by the reduction of the need for assistants in student use sites. The network software is set up to connect the files needed for particular applications; this means that as soon as the student is familiar with starting the computer, the help of an assistant is no longer needed. No disks need to be signed out for most applications, so there is less wear and tear on disks, disk drives, and assistants.

Additional cost savings can be realized as a result of the availability of up to four virtual network drives in addition to the drives contained by the microcomputer. This allows network installations to use PCs without disk drives, a considerable savings in itself. The additional software security provided by such an arrangement can permit a great deal of flexibility in the operation of the network.

Our classrooms in the College of Engineering are designed to be easily accessible to students and faculty. During the normal work week and some evenings, assistants are on duty to sign out manuals and other small items from a locked filing cabinet and to supervise the site, but in practice the classrooms are open to users around the clock every day. Working with the University locksmith, we devised a combination of brackets, bolts, and padlocks that could be adapted to provide security for almost all of the equipment in our classrooms.

INSTALLATION--THE HARDWARE

Long before the installation of a network begins, the groundwork must be laid with detailed plans for site preparation, as it would be for any microcomputer installation. Air conditioning, power, lighting, furnishings, and the physical setup of the classroom must be carefully planned in advance.

The bus configuration of the Ethernet network allows a relatively painless installation, even for novices. A single length of thin Ethernet cable, technically known as RG58, is laid in a path that will allow it to reach each computer on the network, as illustrated in figure 1. Both ends of the bus have reflectors to terminate the network. Each computer is connected to the cable through a T-connector, as shown in the box in figure 1. Several types of connectors are available, but we have found crimped connections the most satisfactory.

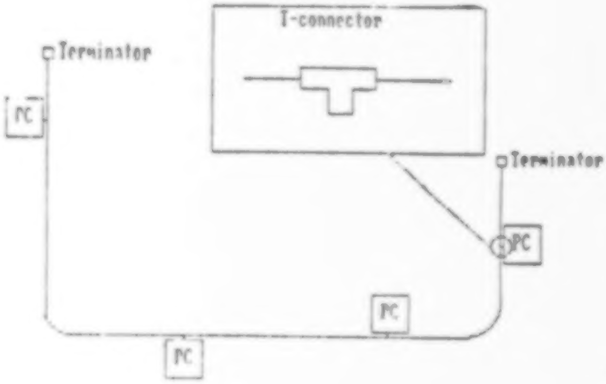


Figure 1. Diagram of network configuration with box showing T-connector.

The length of cable between any two computers connected with thin Ethernet must be at least one meter, and the total length of each segment must be less than three hundred meters. Greater distances can be covered by using repeaters to add segments. Up to ten cable segments can be combined in one network using repeaters. The maximum distance allowed between any two computers connected by thin Ethernet cable is nine hundred meters. Standard Ethernet cable allows a maximum segment length of one thousand meters and a maximum distance between two computers of three thousand meters. Although standard Ethernet cable allows longer networks, thin Ethernet offers advantages of easier installation and lower cost.

Cabling should be largely completed before site preparation is finished, and should be planned in conjunction with electrical wiring. In one of our sites, wiring and cabling were installed in two separate, shielded tracks of the same conduit attached to the walls adjoining the work stations. In all of our networks, dropped ceilings in the buildings reduced considerably the work of extending the cable to individual offices, and utility conduits between floors offered ample flexibility for our uses.

Two methods of purchasing cabling are available. Cables can be purchased made-to-order in the lengths required, but in this case very accurate measurement are required in advance, and errors are rather costly. We chose to purchase large lengths of cable so that we could cut it to the proper lengths on the site. Once the cable is installed within walls and ceilings, it is difficult to accurately gauge the length of a network segment. By dedicating one three-hundred-meter spool of cable to each network segment we were able to avoid the possibility of inadvertently exceeding the maximum length.

NETWORK LIMITATIONS

In theory, an Ethernet installation permits the connection of up to one hundred computers per segment. In practice, with just one computer acting as both a file server and a print server, only about twenty stations can use the network simultaneously without noticeable degradation of response. This number is reduced further if many heavy data transfer operations, such as the compiling of code, are being performed. Printer response is very slow under these conditions as well, and once the printing process has begun, other operations are noticeably slower. These problems of slow response can be considerably relieved, but not eliminated, by the configuration of additional PCs to be used as print servers or file servers.

FUTURE PLANS

The Office of Computer-Based Instruction expects to install more networks on the University of Delaware campus as new software becomes available for network applications. In addition to obtaining commercial software that takes advantage of the network environment, a number of our lessons will be programmed to take advantage of the network to do student data collection. Although it is hoped that 3COM will provide additional Ethernet functions, plans are being made to supplement the network software with code that will address particular needs for our classrooms.

Further plans include linking the present networks to the larger campus Ethernet network that includes the VAX computers in the College of Engineering. The long-range goal is the establishment of a campus-wide network that would allow each PC to function in three ways: as a stand-alone microcomputer, as part of the local network, and as a terminal to the University's mainframe computers.

CONCLUSION

The decision to implement a microcomputer network for an educational setting must be based on a variety of considerations, depending on the function of the network and the goals envisioned for it. Once the decision to implement a microcomputer network is made, the process of building an operational microcomputer network--including selection, cabling, configuration, installation, and security--is certainly within the abilities of novices in network architecture.

WRITING and PRODUCING EFFECTIVE MICROCOMPUTER USER GUIDES

Chris Scherer, Associate Professor

of Communications, University of Illinois

ABSTRACT

As we move into the microcomputer era the challenge before us as educators are varied. One of our roles in the development and use of microcomputer software is to write and produce user guides that are clear, understandable and easy to use.

We must become involved in this type of communication activity or much of the effort of the subject-matter specialists in creating programs will be lost to confusion, misunderstanding and a lack of knowledge about, first, how to communicate this new technology to end users and, secondly, the availabiilty of university-created software.

In this paper, I describe the steps I go through to produce a user's guide including planning, audience analysis, organization, writing style, screen design and some production tips.

While programmers are experts in program design and writing, they are usually not expert communicators. And subject matter specialists or the developers are usually not totally computer oriented. This shortcoming often results in programs that are unnecessarily difficult to operate. I have found that the writer, with expertise in communication, can often make suggestions to improve a program's ease of use, adding to its usefulness and success. Cooperative planning ensures that what the user learns from the user guide corresponds exactly to what the software will do.

As in the development of any message, the audience for the software package must be understood and the message targeted. Usually, the subject matter developer has a good idea who will use the product. Therefore, the more carefully the target audience is defined, the easier it is to tailor the user guide to the needs of the intended audience. The content of the user's guide must be written from the users perspective.

INTRODUCTION

As we move into the microcomputer era the challenge before us as educators are varied. One of our roles in the development and use of microcomputer software is to write and produce user guides that are clear, understandable and easy to use. I have recently assumed the responsibilities of writing and editing user guides and developing marketing materials for software packages being developed by subject matter specialists in the College of Agriculture at the University of Illinois.

I have found the new assignment both stimulating and frustrating. But I am convinced communications specialists must become involved in this type of communication activity or much of the effort of the subject matter specialists will be lost to confusion, misunderstanding and a lack of knowledge about, first, how to communicate this new technology to end users and, secondly, the availablilty of university-created software.

In this paper, I will describe the steps I go through to produce a user's guide including planning, audience analysis, organization, writing style, screen design and some production tips.

Assuming the user has some background in computer use and terminology is a fatal mistake made in some of the materials I have tried to use. And I am sure I am not alone in this. I know of no industry that uses more jargon than the computer industry. And if one of our tasks as communicators is to avoid jargon, we have a huge task before us.

I have found that in writing a user guide, or even editing one the author of the program has put together, I must first thoroughly understand how the program works to be able to operate the program completely. To do this, I sometimes spend a couple of days just playing with a program. Many times during this "play" time, I find a bug or problem with the program. And sometimes I do "the unexpected" thing that the programmer or developer did not anticipate a user would do and I crash the program or at least the program doesn't work as it is supposed to operate. There was a good case of jargon in the use of the word "crash". And for those of us who understand crash it is a good term. But getting back to the topic, I have found that I am not only writing a user guide for the end user but I am also field-testing the product. While working through some programs, I find the screens are either confusing or too crowded. At that point I become a screen design expert. I will discuss some of the techniques of screen design later.

Let me say here, I am not a programmer and more importantly, I do not think I have to be a programmer, nor do I want to be. But I have found it is very important to understand what I am doing and to be able to describe to a programmer of a software package what I was doing when a problem occurred. I try to recreate the problem before going back to the programmer so I can precisely describe when the problem occurred and how it occurred. This of course helps the programmer know where to look for the problem. And just as programmers learn from working on different programs, I have found that as I learn to operate and test new programs, I find new ways to "create problems."

But because no program is perfect, all we can do as educators is to put together the best possible user materials and hope the end user can understand or will seek help when a product fails to perform as he expects it to.

At this point I would like to share with you some tips I have for writing and producing user guides and support materials.

PLANNING

Planning, at its best, is a cooperative effort among many people.

While programmers are experts in program design and writing, they are usually not expert communicators. And subject matter specialists or the developers are usually not totally computer oriented. This shortcoming often results in programs that are unnecessarily difficult to operate. I have found that the writer, with expertise in communication, can often make suggestions to improve a program's ease of use, adding to its usefulness and success. Cooperative planning ensures that what the user learns from the user guide corresponds exactly to what the software will do.

Early planning allows the guide writer to begin writing earlier in the developmental stage and helps to get the software package available sooner. We have always complained about not being included early enough in the developmental process of a program, and the creation of a software package is no different. In fact, those programs which are entirely written before I get to look at them usually take me more time to learn and thus to test and write the guide.

Cooperative planning allows the guide writer to help establish the objectives and the way they will be implemented. I dislike having format, deadlines and budget forced on me. And I feel that when I am involved and have responsibility for the success of a user guide I do a better job.

But perhaps the strongest recommendation I can make for planning is that it facilitates communication between the developer, the programmer and me. There is no way I could produce a user guide in a vacuum. I must communicate regularly with the subject matter developer and the programmer.

Some of the goals of creating a user guide are to enhance the usefulness of the product, explain thoroughly the operations and facilitate the marketing of the product.

AUDIENCE

As in the development of any mesage, the audience for the software package must be understood and the message targeted. Usually, the subject matter developer has a good idea who will use the product.

Therefore, the more carefully the target audience is defined, the easier it is to tailor the user guide to the needs of the intended audience. The content of the user's guide must be written from the user's perspective. For example, I like to use direct commands such as "after you have completed thus and so and the following information appears on the screen, type this." I avoid using quote marks because many novice users think they are supposed to type in the quote marks. Therefore, I like to set what the user actually types in off on a separate line or at least set it in bold face type. I only use italics for material that appears on the screen. But that is getting into style which I will cover a bit later in this paper.

The second important thing to keep in mind concerning the audience is what topics does the user need covered? For example, is a glossary of terms needed? Is a function key page needed? Are some cautions about how to interpret the results necessary? Is additional subject matter needed for clearer understanding?

Perhaps one of the most difficult decisions to make about an intended audience is how much does the user know about running any program. A beginner user needs to know exactly how to make a backup diskette as well as how to install the disk operating system (DOS) on the program disk.

But a more experienced user is bored with too much detail. For the more experienced user, it is important that the user guide be developed so that he can skip around in the material and read only what he needs to move on. I have found that a Table of Contents is essential for this purpose.

Because I began my current assignment as somewhat of a novice, I have tended to write for the beginner. But as I have progressed in the job, I find I get more brief in my explanations. Maybe I am writing tighter copy.

ORGANIZATION

There are many ways a user guide can be organized. But because organization provides structure, it is imperative that one establish a format that is easy to follow. I also have found that because we are providing each county Cooperative Extension Service office in their state with all the programs we produce, it is essential that the organization of each user guide be similar to all others coming out from our operation.

Items I recommend be included in any user guide are:

Title Page
Copyright Page
Table of Contents
What the Program Will Do
Who the Program Is For
Program Developers
Computer Resources Required
Getting Started
How the Program Operates

(This may be done as an expanded example or by working through the various sections. If the program has several major functions it performs, I will make each major function a separate section so the user can skip to each function separately.)

Reference Section
How to Get Help
Appendix
Index
Quick Reference Card

The reference section may include interpretation of results or an explanation of subject matter used in the program.

The appendix often includes worksheets, maps and data input forms.

The index may be of two kinds: one that indexes the program as it runs on the computer so a user knows where to find something quickly in the program or a user guide index to locate information in the user guide. As our programs get longer and more complicated, I am convinced we will have to develop more indexes.

And the quick reference cards or templates are most important for more complicated programs and programs that use special keys such as the function keys or cursor pad keys.

Table of content pages may be only the major topics, or if no index exists, should include subtopics under major topics to facilitate a user's moving around in the guide to find solutions to problems.

WRITING SYTLE

Writing style includes using the right words- short vs. long, concrete vs. abstract, simple vs.. fancy, common vs. jargon, and words rather than symbols, initials and abreviations.

Pace is important in that many readers jump around in a user's guide to find the information they want. Thus it is important to quickly reveal the information you want the reader to get. However, if the material is complex you may decide to slow down the pace with more examples and more complete explanations.

Besides varying the length of sentences, you can pace your writing by paragraph length, headings, illustrations and white space.

I have found that instructional writing should imitate the tone of oral instruction. And one of the best ways to create this tone is to write in the second person using you and your. Avoid saying the "user."

Just as in good journalism style, the use of active voice is important. This is not to say, however, that passive voice should never be used. If the object of the action is more important than the subject, if active voice is awkward, or if you want a sentence to stand out, use passive voice.

Like any good writing you need to develop a style sheet to maintaine consistency. Your style sheet may include such items as abbreviations, capitalizations, jargon, numbers and punctuation. A program I use occasionally to analyse my writing style is Grammatik. This program looks at such things as: sexist terms, word use profile, phrases, sentence length and punctuation.

I spoke earlier of screen design. Although writing for clarity is most critical for producing a good user's guide, screen design can somewhat substitute for writing. Graphics and symbols can be used to carify and punctuate a message's meaning.

Factors I consider when looking at the screens for design include total amount of copy, type style including upper and lower case letters, line length, justification, breaking points and character attributes such as boldface, reverse video, underlining, blinkling or flashing text, and color. Depending on the word processing program, text size, italics and text rotation may be character attributes that also can be used to facilitate readability.

PRODUCTION TIPS

Although it seems quite elementary, I have found that field-testing the user's guide has resulted in subtle changes that can improve a product. For example, one tester of a program recommended the user's guide not be fixed bound because when he opened it up to a particular topic he wanted the guide to remain open without having to weight it down, placing a marker in it or turning over the book to keep his place.

I prepare all my copy using a word processor. I then transmit it electronically to the University Press and provide the Press with a specked hard copy for them to code the specifications. I even have the press put in the screen boxes for examples. Then, all our art department has to do is reduce the actual screen contents and paste them into the boxes in the text.

As a general rule, I begin all new topics at the top of a page regardless of how long the text for the previous page ran. That way, a user who is looking for a particular topic can use the table of contents or index to locate the information needed more efficiently.

A major concern with format has been to leave wide margins so artwork such as keys can be added as illustrations and quick references.

If you have never written a user's guide before, or if this session was a review, I hope I have given you some ideas about writing and producing user giuides for microcomputer programs that are clear, understandable and easy to use.

REFERENCES

Heines, Jesse M. (1984). Screen Design Strategies for Computer-Assisted Instruction, Digital Press, Bedford Mass.

McGehee, Brad M. (1984). Writing Software User Manuals, Writer's Digest Books, Cincinnati, Ohio.

A Comparison Study of Math Attitudes in Programming Classes of PILOT, Logo, and BASIC

Barbara Signer, University of

Massachusetts at Boston

ABSTRACT

The author conducted a preliminary study to investigate observations made while teaching various programming courses to teachers. Inservice teachers in BASIC programming courses often expressed more mathematics anxiety than those in a combination Logo/PILOT programming class. Both were introductory level classes. Subjects from both classes were pre-tested on levels of mathematics anxiety. The treatment consisted of instruction in either BASIC or Logo/PILOT by the same instructor. Upon completion of the courses, subjects were administered the same mathematics anxiety test. Subjects' paired pre and post test scores were tested for significant changes by class. While there was no significant change for the Logo/PILOT group, there was a significant change at the .05 level for the BASIC group. The change was in the direction of an increase of level of mathematics anxiety. The author offers explanations rooted in the way the languages are taught and the constructs inherent in their structures.

In the past, computer courses dealt primarily with BASIC programming. The majority of the students enrolled in these graduate courses were mathematics teachers. With the spread of computer applications across disciplines, teachers from all levels and all areas began to enroll in these courses. By now, most mathematics teachers have already learned how to program in BASIC, through inservice classes and graduate courses. The result is that mathematics teachers today are only a small percentage of the teachers enrolled in beginning BASIC programming classes.

With this change in student population, the author has noticed an accompanying change in "math anxiety". Many of the teachers are verbalizing "fear of mathematics" and relating this to difficulties in BASIC programming. It is common in these classes to hear requests to de-emphasize the mathematics, since many have not taken a mathematics course in a long time or were not "good at math" when mathematics was studied.

Colleges of Education have increased course offerings in computer education. A popular course offered at the University of Houston was a combination of programming in Logo and PILOT. The population was similar to the population enrolled in the BASIC programming course. While both courses emphasized programming the author did not notice similar manifestations of "math anxiety" in the Logo/PILOT group that accompanied the BASIC programming group. In light of this, the author conducted an exploratory study to discern if there were differences in the levels of math anxiety between the two groups. A discussion of the study follows.

BEST COPY AVAILABLE

Methods

The author's purpose was to conduct a preliminary study to investigate if there were differences in math anxiety for teachers enrolled in the following programming classes: a) BASIC, and b) Logo/PILOT. If differences were found, then recommendations for a future study would be made and conjectures given to explain the differences.

Sample

The teacher data source consisted of all teachers enrolled in either the BASIC programming course or the Logo/PILOT programming course offered at the same off-campus site of the University of Houston. Both courses were taught by the same instructor, the author. Twenty-nine teachers were enrolled in the BASIC course and twenty-two were enrolled in the Logo/PILOT course.

Testing

To obtain a measure of mathematics anxiety, the instrument developed by Amburgey (1981) was used. It had been used in a study with pre-service teachers and had a reliability of 0.91. The test was administered on the first and last meeting of each class.

Analysis

The group means of both classes were examined to determine if shifts in levels of mathematics anxiety occured. A repeated measures t-test, pairing subjects' pre-treatment and post-treatment scores, was conducted.

Results

The results include a discussion of a) the differences in the groups' math anxiety when measured prior to the programming instruction; b) the post-treatment group means; and c) the differences in math anxiety when the subjects' pre-treatment and post-treatment scores were paired.

Pre-Treatment and Post-Treatment Group Means

Table 1 contains the pre-treatment and post-treatment means. Lower scores indicate more math anxiety. While both group means indicate similar levels of math anxiety at the beginning of the study, there is a shift in the direction of a higher level of math anxiety for the BASIC group. The Logo/PILOT teachers remained relatively the same.

TABLE 1

MATH ATTITUDE MEANS

	Pretest		Posttest	
BASIC	37	n = 29	33	n = 29
Logo/PILOT	36	n = 22	37	n = 19

Lower scores indicate less math confidence.

Subject Means Paired by Pre-Treatment and Post-Treatment

Table 2 and Table 3 contain the obtained t-value for the subjects' paired pre-treatment and post-treatment scores. Subjects who could not be paired on both a pre-treatment and post-treatment score were eliminated, leaving seventeen subjects in the BASIC group and eleven in the Logo/PILOT group. The results indicate a significant change for the BASIC group but not for the Logo/PILOT group.

TABLE 2

PAIRED t-TEST - Logo/PILOT

	n	Mean	SD	t	Probability
Pretest	11	33	7.508	1.45	.176
Posttest	11	34	5.587		

TABLE 3

PAIRED t-TEST - BASIC

	n	Mean	SD	t	Probability
Pretest	17	37	5.637	2.65	.018
Posttest	17	33	8.470		

Discussion

The results of this study indicate the need for a further study with a larger number of subjects to validate the results of this pilot study. They also cause us to reflect on the way we teach programming. We need to examine the structures of the different programming languages to understand why one can be a more anxious learning experience than another. We must look at the following aspects of the way we teach programming: a) levels of abstraction; b) the nature of assignments; c) student interactions;

d) visual presentations; e) quantitative vs verbal constructs; and f) types of error messages and feedback.

REFERENCES

Amburgey, V.(1981). A measure of pre-service teacher attitudes towards microcomputers. Unpublished manuscript.

Brown, B.R., and Mitzel, H.E. (1966). Post-test questionnaire measuring attitude toward CAI. In H.E. Mitzel & G.L. Brandon (eds), Experimentation with Computer-assisted Instruction in Technical Education. (ERIC Document Reproduction Service.)

Johnson, G.S. (1981). An investigation of selected variables and their effects upon the attitude toward the teaching of elementary school mathematics by prospective elementary school teachers. Unpublished doctoral dissertation, University of Houston.

Willis, J.W., Johnson, D.L., & Dixon, P. N. (1983). Computers, Teaching and Learning. Beaverton: Dilithium Press.

Problem Solving Oriented CAI in University Statics Courses

Dr. Helen Kuznetsov - University of

Illinois at Urbana-Champaign

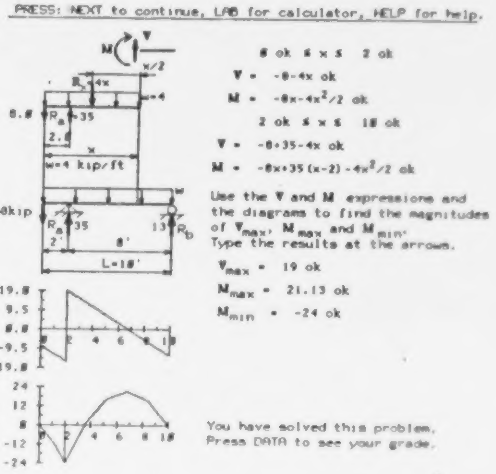
ABSTRACT

The paper presents analysis of development, implementation and students' evaluation of CAI in Statics. The CAI is designed so that each student has to solve an individualized real problem to show his understanding of the theory learned in the instructional part. Sophisticated judgement procedures allow students to write equations and algebraic expressions. Design features and grading system enhance mastery.

Statics is a basic engineering course mandatory for all engineering students. Learning how to apply basic scientific principles to actual engineering problems can be done only by practicing problem solving, the more the better. In this environment, it would be a welcome relief for everybody involved to have an intelligent CAI for problem solving tutoring. To create such courseware was my goal when I started developing PLATO courseware in Statics.

Basic design features of CAI in Statics

1. Each tutorial message no longer than one display is followed by at least a question that reveals the student's understanding of the topic. As soon as it is possible to show some practical application of the theory explained, a short example is presented and then the student has to solve an exercise problem. This approach enhances the learning process, and allows the student to evaluate his understanding of the topic.
2. Each question has a relevant help message available to the student through the HELP key. The help message may contain remedial information in the areas usually studied in other courses, for example, trigonometry or vector algebra formulae from mathematics. In this way the remedial information is presented only in the case when the student needs it, without wasting the time of the students who are fluent in mathematics.
3. Each question has at least one response to a predictable wrong answer. Of course, this type of design can be developed only by a person who had substantial experience in teaching the subject matter.
4. After three unsuccessful attempts to answer a question, the student gets the correct answer which he has to type in anyway. Typing in the correct answer enhances the learning process. There is no frustration associated with not being able to move in the lesson. When the response is graded, the number of points for the answer is drastically reduced each time the student sees the answer. Of course, this feature can be incorporated only when the student gets a completely different problem each time he enters the test.
5. All CAI lessons in Statics are designed so that the student can freely move in the lesson back and forth. The student may return to the explanation he wants to review but he or she is never forced to do so.
6. All the lessons have the test in the end of the lesson which is graded. This feature allows the instructor to use the CAI as a home work assignment or a quiz. Each time the student enters the test, he or she gets a different problem. In some lessons the difference is only in the numerical data, where in the other lessons the entire problem including the sketch is different.



PRESS: NEXT to continue, LAB for calculator, HELP for help.

0 ok $\le x \le$ 2 ok

$V = -8-4x$ ok

$M = -8x-4x^2/2$ ok

2 ok $\le x \le$ 10 ok

$V = -8+35-4x$ ok

$M = -8x+35(x-2)-4x^2/2$ ok

Use the V and M expressions and the diagrams to find the magnitudes of V_{max}, M_{max} and M_{min}. Type the results at the arrows.

V_{max} = 19 ok

M_{max} = 21.13 ok

M_{min} = -24 ok

You have solved this problem. Press DATA to see your grade.

Fig. 1 Shear Force and Bending Moment Diagrams Test

7. To enhance learning, the student is allowed to return to the test and to improve his or her grade. This feature was added on the students' request just recently. The result was somewhat unexpected: all the students in the group (there were 14 students) worked in the lesson until they got a 100% grade. However, some spent just 40 minutes to master the topic, when others spent 2 hours to reach the same goal. This was lesson "Beams under Uniformly Distributed Loads," where the test requires the student to solve a simply supported beam loaded with a force and a distributed load on a

BEST COPY AVAILABLE

portion of a beam. The solution includes writing equilibrium equations to solve for unknown reactions, calculating the reactions, then writing the analytical expressions for the shear force and bending moment on each portion of the beam, and, finally, determining the maximum and minimum bending moment and the maximum shear force. This was one of the homework assignments in a Statics course in the School of Architecture in the Spring Semester, 1985.

8. Wherever the problem can be solved in several different ways, the student can choose the method of solution. One of the choices is to type in the final answer.

Students' evaluation of PLATO lessons in Statics

The students' performance was closely monitored. It was done in two different ways: through the data file and by attaching a questionnaire for students evaluation to each lesson. There were five multiple choice questions and one open ended question where each student was asked to indicate what he or she liked or disliked in this lesson.

The student responses were read every day and changes were made when it was necessary. For example, some students had asked to give an example whenever there was a request to write an expression. After the statement "Write the expression for a shear force" or "Write equilibrium equation $\Sigma M_a = 0$," examples of expressions were presented in the lessons.

Following are some of the remarks that were made right after the student had completed the lesson.

General positive remarks:

"I feel these lessons would be very valuable to have all along the way in a statics or strength of material class. They explain clearly and are fun to do."

"I liked the lecture type material that a student could use to review or when they missed a class"

"The best features of Plato are the graphics and the optional explanations"

"Excellent"

"I enjoyed the lesson"

"I enjoyed working on Plato and I learned the lesson"

"They (the lessons) are appropriate for our class work."

"Immediate feedback pertaining to a correct or incorrect answer has been proven to be the best method for learning any and all things"

"The problems help you learn through experience."

Specific likes:

"Liked: the progression of the magnitude of difficulty, the help format"

"I like the step by step lessons and clear explanations"

"I like the capability of working problems unaided"

"I like slow examples"

"I like being able to work at my own pace"

"I like the three degrees of problems"

"Lesson very helpful for comprehending concepts"

"I like the fact that I can get help quickly"

Remarks about graphics:

"I like the bending moment diagrams"

"Seeing the deflected shapes was helpful"

"Diagrams are helpful (more than in lectures)"

"Graphics for learning right hand rule were great and pertinent problems following each new idea were also helpful."

Comments on implementation by the instructor:

"I wish my teacher would have given me the PLATO exercises before the quizes when they would have helped our grade!"

"I like the sample problems explained fully like they are before the quiz."

Conclusions

I. It is better to develop topic packages instead of the course geared to faculty replacement with a computer because of the following reasons:

1. Topic packages can be easily included in any appropriate course. The instructor may make his or her own decision whether, when, where and how to incorporate the topic package in the course. The topic package may be recommended to students for remedial purposes, or as a supplementary material for study on their own, or for preparation for professional examinations. Ideally, there will be a CAI library of available topic packages.

2. Theoretical courses such as mathematics or physics may use some of the problem solving lessons as an illustration of the application of the theory they teach.

3. In general, as in any area of human activities, extremes (replacing completely one type of instruction with the other type) proved not to be beneficial for everybody involved. For example, text books did not replace lectures.

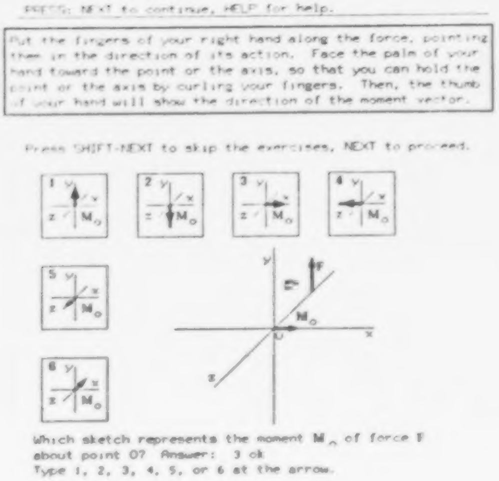
II. CAI should not be compared with lectures or practical exercises in class on the presumption that either of the instructional techniques is better or worse than the other one, but rather each should be incorporated in the teaching process naturally where it serves better than any other media.

III. Suggested areas of instruction where CAI proves to be more efficient than lectures or classroom tutorials:

1. Problem solving exercises where properly designed CAI provides immediate response to the student answer, giving the additional information only when necessary.

2. Homework assignments dealing with problem solving where the student gets a score calculated by the computer on the basis of his/her performance. To enhance learning, the CAI design may allow the student to improve his/her grade by returning back and solving another problem.

3. Drills on new rules or sign conventions.



PRESS: NEXT to continue, HELP for help.

Put the fingers of your right hand along the force, pointing them in the direction of its action. Face the palm of your hand toward the point or the axis, so that you can hold the point or the axis by curling your fingers. Then, the thumb of your hand will show the direction of the moment vector.

Press SHIFT-NEXT to skip the exercises, NEXT to proceed.

Which sketch represents the moment M_o of force F about point O? Answer: 3 ok

Type 1, 2, 3, 4, 5, or 6 at the arrow.

Fig. 2 Drill display for the right hand rule.

4. Sophisticated theory or method of problem solving illustrated by graphics. Computer graphics techniques allow the lesson designer to keep on the screen only the part of the sketch which is explained in the text on the same display. When new explanations appear all the details not refered to in this text may simply disappear from the screen and new details may appear in the sketch.

IV. Educational strategy that works the best in engineering education is to require the student to apply every bit of knowledge he obtains to solving some practical problems, and to do so right away, the sooner the better.

If the understanding of the topic is checked by multiple choice, true-or-false questions, or similar instructional strategy, the passing grade says nothing about the student's ability to use this knowledge. The only proof of the student's understanding of an engineering subject is the ability to employ the theory for solving practical problems.

If the CAI is developed mostly by educators without an engineering background, the crucial importance of the problem solving ability may be missed, and the CAI developed will be rejected by the engineering faculty.

BEST COPY AVAIL

Unfortunately, this is often the case.

V. The worst educational strategy for CAI is to force the student to review some parts of the lesson after failure to answer some questions. In 90% of the cases I have observed, the answer judged as wrong was not really wrong. There are many reasons for this type of judgement failure:

1. The question is not stated clearly
2. The judgement procedure is not sophisticated enough
3. This type of correct response was not anticipated by the author
4. The student is not familiar with the computer and the keyboard
5. There is a mistake in the program

VI. Sometimes the judgement of CAI applicability in engineering education is made on the basis of bad experiences with inappropriately designed lessons. Somehow computers are blamed for the authors fault, and conclusions are made that it is not possible to present a real problem in CAI because of computers limitations.

VII. The best way to design a test problem is to have a computer generated individualized problem for each student.

Statistical data

The work was sponsored by the Computer Education Research Laboratory at the University of Illinois, Urbana-Champaign, Director, Professor Donald Bitzer.

Statistics on Usage of PLATO Lessons

Semester	Number of Students	Number of Lessons SM	HA*	Average Time on PLATO per student	Want PLATO Lessons
Fall 83	22	3		2.0 hours	95%
Spring 84	39	5		6.2 hours	65%
Fall 84	40	2	4	8.5 hours	80%
Spring 85	14	1	7	8.4 hours	50%

* SM - supplementary material HA - homework assignment

In the Fall Semester 1984 I was teaching a Statics course in the School of Architecture, and collected the following data.

Staticstics on Distribution of
Students' Usage of PLATO Lessons
Among the Students Who Received
Different Final Grades

Grade	PLATO time per semester Min.	Max.	Ave.	Number of students
A	4.4	19.3	10.3	12
B	2.7	11.9	7.2	12
C	3.5	16.2	8.5	12
D	5.4	5.4	5.4	1

References

Kuznetsov, Helen. (1983). Moment of a Force - Basic Concepts [Computer-based instructional program]. Urbana, IL: University of Illinois, Computer-based Education Research Laboratory (CERL), (Producer). Minneapolis, MN: Control Data Corporation (Publisher).

Kuznetsov, Helen. (1983). Moment of a Force - Space Problems [Computer-based instructional program]. Urbana, IL: University of Illinois, CERL, (Producer). Minneapolis, MN: Control Data Corporation (Publisher).

Kuznetsov, Helen. (1984). Shear Force and Bending Moment Diagrams 1 [Computer-based instructional program]. Urbana, IL: University of Illinois, CERL.

Kuznetsov, Helen. (1984). Shear Force and Bending Moment Diagrams 2 [Computer-based instructional program]. Urbana, IL: University of Illinois, CERL.

Kuznetsov, Helen. (1983). Bending [Computer-based instructional program]. Urbana, IL: University of Illinois, CERL.

Kuznetsov, Helen. (1983). Quiz on Combination and Resolution of Forces [Computer-based instructional program]. Urbana, IL: University of Illinois, CERL.

Kuznetsov, Helen. (1983). Force in Space [Computer-based instructional program]. Urbana, IL: University of Illinois, CERL.

Kuznetsov, Helen. (1982). Statics - Combination and Resolution of Forces [Computer-based instructional program]. Urbana, IL: University of Illinois, CERL.

Note: On CDC systems the programs may be accessed through the "Authors' Library" (by typing "2authors" on the author mode page). On the CERL system the programs are available through the on-line catalog (by typing "uicat" on the author mode page).

PLATO systems may be accessed from microcomputers by using the PLATO access disks distributed by Control Data Corporation, CERL, and other vendors.

A VIDEOTEX IN YOUR FUTURE?

Elisabeth M. Craig

University of Tennessee at Chattanooga

ABSTRACT

The experiences of our department in implementing and utilizing the comparatively new videotex technology are described. Included is a description of the system and its facilities, as well as a discussion of the linkage to the host computer. Considerations in designing screens (color, shapes and text) are reviewed. The paper provides an overview of some educational projects for which videotex is a viable choice.

INTRODUCTION

Suppose you were lucky enough, as we were, to be awarded a videotex system, how would you integrate it into your program? We believe that we have made a good start in answering this question. Some of our experiences and ideas may help you with such a system.

In case you are not familiar with videotex, it is a generic term coined by the CCITT (International Telephone and Telegraph Consultative Committee) and known under various names such as Prestel (England), Teletel (France) or Telidon (Canada). The background of videotex is well articulated by Roger Woolfe (Woolfe, 1979). Imagine a screen with high color resolution which allows you to design a series of frames displaying graphic and text information. Many shopping malls, motels and state tourist centers are equipped with these devices which show, in colorful format, local attractions and pertinent information for the traveller.

The old adage that a picture is worth a thousand words is even more true when the "picture" has the flexibility and high resolution color of videotex. Equally important is the fact that the displays can be modified quickly to respond to the needs of specific audiences.

What kind of resources do we have to support this system? In 1984 the State of Tennessee provided funds allowing UTC to establish a Center of Excellence for Computer Applications (CECA). The Center is equipped with an IBM 4381 system and peripheral equipment. In addition, UTC has two Hewlett-Packard 3000 systems. One of the objectives of this Center is to promote state-of-the-art technology to our faculty, students and staff. The AT&T Frame Creation videotex system, awarded in November 1985, with its vast range of possibilities, was one of the first enhancements to the Center.

DESCRIPTION AND USE

The AT&T videotex system can be used either as a stand-alone (off-line) unit or as an on-line peripheral to a host mini or main frame. As a stand-alone the system stores material on 8 inch floppy discs, which also are the medium supplied by AT&T to start up the system.

The videotex system is equipped with a plotter bed and stylus. Ten geometric shapes (circles, rectangles, arcs, points and lines) are incorporated into the system. These pre-designed forms provide the basic building blocks for developing more complex graphics. The shapes are displayed in a menu and can be rotated or scaled to create other shapes. In addition, there is a grid which can be used in designing the screen; the grid is "hidden" when the frame has been completed.

There are over 25,000 colors available, but only sixteen at a time can be kept in the menu. The menu is used to manipulate the hue (the color itself) and the chroma (degree of whiteness) of the colors as well as to emphasize the redness or blueness in a color.

Since it was simpler to use the system as an entity by itself, our first sessions used this approach while we learned how to handle the system and plotting pad. However, as we became more familiar with videotex, we needed the on-line link to the host HP3000 Series III, because this allowed storage and retrieval of frames between the central main frame and the various monitors around campus. Off-line we could access only one mode (Frame Creation), but on-line we could utilize the three additional modes (Dialog, File Transfer and Display). We knew that making the connection would not be an easy task. Other educational users of the AT&T videotex in the Southeast were contacted, but none had an HP3000, while only a few were venturing to make the main-frame connection.

We went through a four week trial and error phase in which we first tried the easiest solutions, found they wouldn't work, then attempted more difficult techniques. The techniques involved developing a break-out box to make the physical link, setting the link levels (i.e. the handshake process), and establishing a session level communication. If you would like more complete details on this, feel free to contact us.

Once we had the host HP3000 Series 48 and the videotex talking to each other,we had to supply the dialog (reword that to software) so that we could take advantage of the different modes in videotex and also store data and information on HP disk packs.

We decided the first application for our videotex would be to create screens indicating areas on the campus designated for student parking. The application was not only useful to UTC users, but the map could be reused for other applications, so we could learn and be productive at the same time.

We started with a map of the campus showing the two types of parking available. Subsequent frames enlarged parts of the map so that specific areas could be displayed. The complete set of frames was linked in a logical sequence taking a street by street route around the entire campus. The frames are displayed in a repeating cycle which changes at a rate of two frames a minute, (the rate can be varied from .1 to 30 seconds per frame.)

In the process of designing the application, a number of factors had to be considered. First, if other departments intend to use videotex to display information etc., the level of software must be developed accordingly. The software must be very user friendly and easy to use since, in many cases, the people entering the data will have limited knowledge of software. If they find the system difficult to use, they will "forget" to update the information and your target audience will soon disappear. For example, if you use videotex to display frames describing campus activities, but these are out of date, the students will ignore the monitors thereby undermining the videotex project.

There are several practical rules for designing screens or displays. These include easily readable text, variety of layout, consistency of text with graphics and the subdivision of complex material (Kosel and Jostad 1983). There are other rules we would add, such as keeping the screen simple, using as few colors as possible, and keeping contrast colors, such as very dark or very light, close to each other. Other authors note that "if the goal is to convey text or graphic information, opponent-color combinations should always be avoided. Yellow on a blue field and red on a green field produce the sensation of 'shadows' on the display" (Durrett and Trezona, 1982). Baker points out that colors represented by similar patterns should be put some distance from each other (Baker, 1983). We found that while a blinking color may attract the eye to the first screen, on subsequent screens it can be distracting. In an extensive article, Baldwin discusses other factors involved in human color perception and how this knowledge should be used in designing graphics (Baldwin, 1984).

If you intend to translate a videotex screen into a slide (this can be done with a slide creator), there is another factor, namely that a frame which is attractive on a screen may be lost when projected. This requires careful consideration of the size of text, background and contrast colors. Also, it has been noted by researchers that graphic formats are superior to text formats for training or instructional purposes (Tullis, 1981). The text on videotex frame can be varied in width from one column width to forty columns and can be rotated in increments of 90 degrees. Careful consideration should be given to the width of the text so that it is easily legible.

Looking to future applications, we plan three separate project areas, each fostering applications based on the Frame Creation System. These can be broadly defined as (1) CECA orientated (2) administrative and (3) student orientated.

Graphic displays of the facilities, capabilities and projects associated with CECA will provide UTC with a powerful method for making the students and the community aware of the Center. Our next project will walk students through the process of signing onto the computer systems and show them how to use specific programs or equipment. Another project will provide frames giving information on CECA, the location of its equipment and terminals.

Computerized color graphics including flat-bed plotters, high resolution CRTs, along with appropriate software will be linked to the videotex system as they are acquired. This will support graphics research within the Computer Science and Engineering departments.

Administrative areas can benefit from a videotex facility linked to the UTC student registration system (or STARS) which is based on direct computer input of student schedule information. Videotex can improve the efficiency of this system by cycling frames telling the students which sections are closed and which are new offerings besides denoting room changes. The frames will be color coded by major allowing for quick identification by the students.

If all goes well, we expect to have trial frames ready for the Fall '85 semester.

The Admissions Office can use videotex to design vivid, attractive advertising messages providing information on various departments, majors, etc.

The Engineering department is expected to adapt videotex to an existing monitor system that delivers messages, information, and announcements to the students on their way to class. Students and faculty in both undergraduate and graduate programs can achieve proficiency in videotex computer applications to support their research activities and professional development.

Since Videotex offers strong visual orientation in many areas of academic and public presentation of information, this graphic form can improve tutorials in many areas: Computer Science, Theater, Art, Communications, Physics, Engineering, Math, Biology, English, and other disciplines. Typical applications might be earthquake wave simulations (Geology), advertising and orientation (Military Science), macroeconomics modelling (Economics), demographic maps (Sociology) and optical illusions, physiology and psychological testing (Psychology).

Dr. Eric Lane, author of software distributed nationally by Conduit, sees many possible applications in his area including "the design and implementation of a multi-level tutorial facility in introductory physics for use in high schools and in introductory college and university physics courses" (Lane, 1984).

In Fall '85 we will be offering workshops and short courses to our faculty and staff to acquaint them with the videotex equipment. We have fifteen terminals which can be loaned to individuals or to departments. We anticipate that this kind of support will encourage faculty to develop computer assisted instruction, information displays and, most important of all, to devise new ways of using the technology. Their experiences and ideas will in turn provide us with valuable feedback. During the process of contacting other videotex users in the Southeast, there was much interest shown in developing a network to share ideas, problems and techniques.

Tennessee has, as a top priority, integrating its industry and schools with high-technology. To attain this, Tennessee has introduced a program for computer literacy in grades K - 12, has instituted data processing courses in high schools, and has inaugurated training programs in areas such as computer science, electronics, and robotics. In this type of innovative climate, videotex has a viable place as a new tool for instructional technology at all levels of education.

Based on six months experience, we look forward to the development of diverse applications and to making greater and more creative uses of this system. One of the exciting aspects of videotex is that it is flexible and versatile, so that it can be adapted to a wide variety of disciplines and student age groups. It encourages the user to explore, offers the capability of investigating a multitude of shape and color combinations and is an attractive medium with which to work. If you have a videotex, enjoy!

REFERENCES

Baker, Patti. Computer Color Graphics and Monochromatic Display, Educational Technology, 1983, 23(12), 22-24.

Baldwin, Lee. Color Considerations, BYTE, 1984, 9(9), 227-247.

Durrett, John and Trezona, Judi. How to Use Color Displays Effectively, Byte, 1982, 7(4) 50-53.

Kosel, Marge and Jostad, Karen. Designing the Display, Pipeline, 1983, 8, 29-32.

Lane, Eric. Proposal submitted to AT&T, UTC, 1984.

Tullis, Thomas S. An Evaluation of Alphanumeric, Graphic, and Color Information Displays. Human Factors, 1981, 23(5), 541-550.

Woolfe, Roger. Videotex: Butler, Cox and Partners, Ltd, London, 1979).

SIMPLE INTERACTIVE VIDEO DISC EXPERIMENTS ON MOTION IN PHYSICS

Ronald Gautreau

New Jersey Insititute of Technology

ABSTRACT

One of the beautiful features of Interactive Video Discs (IVDs) is that their structure makes them readily suitable for use in motion analysis. In this paper we describe various IVD experiments illustrating fundamental physical concepts that are used in Physics laboratories at the New Jersey Institute of Technology (NJIT). The experiments can be performed using only a basic Level 1 videodisc player, or a more sophisticated Level 3 system such as Digital Equipment Corporation's IVIS system.

LEVEL 1 DISTANCE AND TIME MEASUREMENTS

An important concept in the study of motion is the understanding of how the position of an object varies with time. An IVD is especially suited for motion analysis. The frame number indicator serves as a built in clock, since the frames are separated from each other by 1/30 second. Further, the numbers are also convenient to use, since every 3 frames corresponds to a time passage of 0.1 second.

There is an especially simple procedure that we have used for measuring the distance of a moving object on an IVD. We made a transparancy of a sheet of ordinary graph paper. We then placed this tranparancy grid over the screen of the TV monitor. The transparancy stayed in place on the screen because of static electricity. This gave us a simple means of overlaying a position grid on top of our TV picture at essentially zero cost. The grid markings were calibrated to read distance by comparing the markings with some known distance on the TV picture.

Thus, the combination of an overlayed calibrated grid together with the IVD frame number indicator provides a basic setup for any Level 1 video disc player to be used for analyzing motion sequences displayed on the IVD screen. By using the step function control of the video disc player, a student can step a moving object through its motion, record its grid marking and corresponding frame number, and thereby measure the object's position as a function of time.

The Physics IVD laboratories that we have performed at NJIT are built around motion sequences that exist on two commercially available level 1 video discs, "SpaceDisc 5--Astronomy," Video Vision Associates, and "The Physics of Automobile Collisions" by Dean Zollman and Robert Fuller, John Wiley Publishing Company.

MEASUREMENT OF ACCELERATION

The "Astronomy" video disc has three successive chapters showing: a) a person dropping a hammer and a feather on the earth; b) an astronaut dropping a hammer and a feather on the moon; and c) objects undergoing accelerated motion in an accelerated space ship. By using the stop frame technique discussed above to measure distance vs time, the students were able to determine the acceleration of the bodies for each of the above situations in the following manner. For a body starting from rest, its distance s as a function of time t is related to its acceleration a by

(1) $s = \frac{1}{2}at^2$.

If both sides of (1) are divided by t, one obtains

(2) $s/t = \frac{1}{2}at$.

Thus, when the students plotted their measured values of s/t vs t, the result was a straight line whose slope equaled a/2.

With this procedure, the students were able to use the video disc sequences to measure the acceleration of gravity of a body on the surfaces of the earth and moon, and also to measure the acceleration of an object released in the accelerating space ship, comparing their results with the values given by the astronauts.

MOMENTUM IN AUTOMOBILE COLLISIONS

The "Physics of Automobile Collisions" disc has sequences showing various types of automobile collisions in normal and slow motion that are especially well suited for illustrating the concepts of impulse and momentum. The laboratories involved measuring the distance versus time of the moving automobiles, and the collision time.

A fundamental principle of Physics is that the impulse given to a body will be equal to the change of the body's momentum. If we consider only motion in one direction, a body's momentum is the product of its mass and velocity, momentum = mv, and the impulse given to the body by a force acting during a time interval Δt can be written as impulse = $F_{avg}\Delta t$, where F_{avg} is the average force exerted on the body. The relationship between impulse and momentum is

(3) $F_{avg}\Delta t = mv_f - mv_i$.

where the subscripts indicate the body's initial and final velocity.

One sequence used from "The Physics of Automobile Collisions" disc shows an automobile crashing into and rebounding from a wall. The students measured the grid markings vs frame numbers of the automobile before and after the collision by stepping the automobile through its motion using the step function of the remote video disc controller, as described above. The students also measured the time the automobile was in contact with the wall by using the step function and noting the frame number when the automobile first contacted the wall and the frame number when the automobile lost contact with the wall.

The students plotted the initial and final position vs time, obtaining straight lines whose slopes equaled the initial and final velocities, and computed F_{avg} from (3). They then compared their value of F_{avg} with what might be expected from work-energy considerations.

Another sequence from "The Physics of Automobile Collisions" disc shows two automobiles originally traveling at right angles to each other undergoing a collision, sticking together, and moving off at some angle. This illustrates a standard problem discussed in most elementary Physics texts about conservation of momentum in two dimensions, where momentum is conserved in both the x and y directions according to

(4a) $\sum(mv_x)_i = \sum(mv_x)_f$,

(4b) $\sum(mv_y)_i = \sum(mv_y)_f$.

The students measured the initial velocity of each automobile before the collision according to the method described above. They were thenasked to use (4) to calculate the final velocity and angle relative to the x-direction of the coupled cars after the collision. After doing the calculations, the students checked their answers by measuring the final velocity with a position grid, and the final angle using an overlayed transparancy made from a sheet of angle graph paper.

LEVEL 3 EXPERIMENTS

We are in the process of making the above Level 1 experiments into Level 3 interactive experiments. At Level 3, the motion of the bodies will be controlled from the associated computer keyboard instead of the remote control of the video disc player. The computer will also generate overlays on the video disc picture of a clock showing elapsed time from when the student starts measuring time, and a distance grid calibrated by the student.

As the student makes his distance and time measurements, he will enter the data directly into the computer. When the experiment is completed, the computer will perform the analysis of the data, plotting the graphs from the student's data, determining the slopes of lines, etc. The computer will also generate overlayed text explaining the details of each experiment.

The Level 3 development is being done primarily on Digital Equipment Corporation's Interactive Video Information System (IVIS). In addition, the Level 3 interactivity will be extended to other computer systems as well.

DISCUSSION

The motion situations described above are discussed in most elementary Physics text books. However, the TV visuals of real astronauts producing objects in motion, and flying glass and crushing metal in automobile collisions, certainly are more exciting than what can be shown in a text book. Moreover, the analysis done by the students is exactly the same as what would be done by professional scientists or engineers in analyzing the results.

There are various ways the motion data can be analyzed at Level 1. The data can be processed by entering the raw grid markings and frame numbers measured by the students into a computer that has been pre-programmed to convert the grid markings and frame numbers to distances and times using the conversion factors determined by the students, with the computer analyzing and plotting the data. Alternatively, the data can be analyzed by the students themselves using a commercial software package such as TK!Solver. Or, the calculations can simply be done on a hand held calculator, with hand drawn graphs.

The main drawback at the present state of using IVDs in Physics education is the lack of suitable Physics sequences on the video discs that are now available. NJIT is in the process of producing video discs that will be useable in the instruction of Physics at the pre-college and introductory college levels. A summary of the present IVD situation in Physics education can be found in a recent article "The Physics Teacher and the Videodisc," L.D. and D.S. Kirkpatrick, The Physics Teacher p.413, October, 1985.

Evaluating Interactive Videodisc Instruction:
A Procedural Model

Janis Morariu, Ph.D.

Center for Instructional Development
and Evaluation

University of Maryland University College

ABSTRACT

This paper examines the nature of interactive videodisc systems in instruction/training as a new medium rather than as a combination of media formats. A procedural evaluation model is proposed for use in both formative and summative evaluation during design and testing, as well as in the selection of appropriate programs by potential consumers.

INTRODUCTION

With the advent of random-access, high technology media, such as the optical videodisc player and relatively inexpensive microcomputers, industry, government, and education are experiencing an unprecedented growth in the demand for effective and efficient interactive videodisc instruction/training programs (Bowen, 1979; Cerf, 1984). Irrespective of application, the development of interactive videodisc instruction/training programs is not only creating a demand for appropriate in-house formative and summative evaluation tools, but for an evaluation tool to assist outside evaluators in making appropriate program selection decisions for specific applications and audiences.

INTERACTIVE VIDEODISC: MORE THAN THE SUM OF ITS PARTS

An age-old question faced by instructional designers and evaluator's is: how should instruction be presented for optimum learning to take place with the identified audience? Research on learning and cognition shows that each individual brings a variety of previously acquired concepts and skills to any working/training environment (Ausuble, 1963; Bruner, Goodnow, & Austin, 1967). In addition, the manner, or context, in which a task is executed or information is experienced affects human performance (Jenkins, 1974; Watkins & Tulving, 1975). The interactive videodisc would seem to provide a learning environment which combines other media formats in a kind of composite instructional environment. However, as Gavriel Solomon (1979) proposes in his book, Interaction of Media, Cognition, and Learning, each medium is more than the sum of its parts.

Salomon (1979) supports that each media form possesses its own unique set of attributes which, "when used as carriers of the critical information to be learned, call on different sets of mental skills and, by so doing, cater to different learners" (or users)" (p. 9). Not only is the task itself structured through the medium, then, but so too are the thought processes of the user or learner. In the case of interactive videodisc systems, a variety of relevant sound, film, graphic, or text segments in any combination can be accessed to communicate a message. According to Salomon's (1979) theory, the interactive videodisc environment is not serving to combine the various symbol systems of different media formats, but rather, the interactive nature of such environments transforms this combination of individual media formats into a new symbol system requiring new skills, knowledge, and thought processes among users. The overall task of learning from interactive instructional environments, then, is distinct from the sum of its parts. The learning tasks are restructured through this new interactive environment and requires a unique model of evaluation to analyze its effectiveness.

A MODEL FOR EVALUATING INTERACTIVE SYSTEMS

Ben Shneiderman (1984) provides a very useful syntactic/semantic model of the knowledge required by users in comprehending and operating interactive software programs. Very simply, the application of software programs to any utility task (e.g., word processing or spreadsheet accounting) or new learning task (e.g., math, science or emergency medical concepts) requires that the user either already possess or acquire knowledge of the concepts and operations of the problem domain (e.g., writing, accounting, math, emergency medical training), as well as the concepts and operations of the computer system (e.g., branching, cursor movement, help screens, keyboarding, files). Similarly, the process of formative evaluation which occurs during the design and development of a program, as well as the summative evaluation which occurs upon completion of a program, should include an in-depth analysis of: 1) the problem domain concepts and operations, 2) the computer-related concepts and operations, and 3) the methods/techniques through which information is conveyed (Morariu, in press). Adapted from a model for characterizing programmer behavior (Shneiderman & Mayer, 1979), the syntactic/semantic model as applied to the evaluation of interactive videodisc environments proposes that an in-depth evaluation of an interactive instruction/training system include the following three components and related subcomponents:

I. Probem Domain Concepts and Operations:

- Semantic Knowledge of Concepts - a conceptual understanding of the task to be completed or the information that is to be learned (the overall goals, objectives, and concepts or skills of the task). For example, a training program in anatomy & physiology for emergency medical technicians requires a knowledge-base on the systems of the human body, the ways these systems function under normal conditions, and the signs that a system is diseased or damaged.

- Syntactic Knowledge of Operations - a knowledge of the basic problem domain skills and/or concepts for completing a task. For example, the emergency medical technician must know methods/techniques to identify the systems of the body that are damaged, and be able to communicate the information to medical specialists (eg. initial assessment and treatment procedures, and spelling/pronunciation of key information).

II. Computer-related Concepts and Operations:

- Semantic Knowledge of Concepts - a knowledge of how the task is structured by the computer software program. For example, the program may use a tree structure design in which the same information can be accessed from various points/paths through a main-menu/submenu structure. In the program on anatomy & physiology training, the learner can proceed through the program from start to finish, or the learner can choose to view identified segments of the program through a main menu option.

- Syntactic Knowledge of Operations/Functions - an understanding of the specific details of software program operations and functions. For example, to select a set of options to view, the user needs to touch an appropriate area on the screen with a light pen, or type a response and then press ENTER.

III. Information Conveyance:

- Media Formats - effectiveness of audio information, and still and motion visual sequences, including graphics (studio and computer-generated), animation (studio and computer-generated), live film/video footage, and still frame images.

- Screen Display - readibility of text (size, font, and color), placement of text and graphics on the screen, use of color, use of computer-generated masks (eg., blocking out portions of the screen for text overlay).

Guidelines in evaluating information conveyance have been derived from existing research and recommendations (Bork, 1983: Mills & Weldon, 1984; Penna, 1984).

EVALUATION PROCEDURES

Given the syntactic/semantic model for the evaluation of interactive instruction/training systems, the evaluator should complete the following five comprehensive review phases:

- Phase 1: An overall review of the program in order to become familiar with program concepts and operations. This should be completed with a `non-critical eye`, approximating the way in which the learner would approach the program. No notes should be recorded during this phase of the evaluation.

- Phase 2: Review from the point-of-view of the problem domain concepts and operations. During Phase 2 the evaluator should focus on the questions related to the information being conveyed, appropriateness for the audience, and the skills/knowledge required relating to the problem domain.

- Phase 3: Review from the point-of-view of the computer-related concepts and operations. During Phase 3 the evaluator should focus on the questions related to the knowledge and skills required to actually comprehend and operate the computer system.

- Phase 4: Review from the point-of-view of the media format and screen display. During Phase 4 the evaluator should focus on the questions related to the methods through which information is conveyed, as well as the design of the information (eg. text size, style, color; graphics design; technical film/slide quality, etc.).

- Phase 5: Review all notes recorded for Phases 2 - 4 and complete an in-depth constructive critique of the program, including both positive and negative features/characteristics of the program. It is also important to suggest methods of improving any noted negative features.

During Phases 2 - 4 the evaluator should record any and all impressions of the program from each phase perspective, including the problem domain, computer-related concepts/operations, and information conveyance. It is natural during any phase of the evaluation for an evaluator to notice problems related to another phase of the evaluation. It is a good idea to make note of such observations as they occur. However, it is also important for the evaluator to try to maintain the appropriate perspective of the current phase as much as possible in order to be able to complete an in-depth and focused analysis during each phase.

CONCLUSION

The environment created through an interactive videodisc instruction program is greater than the sum of its parts. The interaction of film, audio, still images, and text, as well as the facility for branching, customized immediate feedback, and record-keeping, creates a unique symbol system that should be analyzed from a new perspective. The need for an evaluation model that deals specifically with the unique learning/training environment introduced through interactive videodisc systems is recognized through the proposed five-phase procedural evaluation model. Both an in-house formative and summative evaluation can be conducted using this model. In addition, this model can be used by outside evaluators making program selection decisions for specific applications and audiences.

REFERENCES

Andrews, D. and Goodson, L. (1980) A comparative analysis of models of instructional design. Journal of Instructional development, 3 (4), 2-15.

Ausubel, D. (1963). The Psychology of Meaningful Verbal Learning. New York: Grune & Stratton.

Bork, A. (1983). A preliminary taxonomy of ways of displaying text on screens. Information Design Journal, 3 (3), 206-214.

Bowen, C. (1979) A Survey of Videodisc Applications in Industrial Training and Development. Cambridge, MA: Kalba Bowen Associates, Inc. Bruner, J., Goodnow, J. J., & Austin, G. A. (1967). A Study of Thinking. New York: Science Editions.

Cerf, C. (1984) Exec Learning Company A goose, a boot, and a thing called excellence. Softalk, 4 (1), 60-67.

Jenkins, J. J. (1974). Remember the old theory of memory? Well, forget it. American Psychologist, 29, 785-795.

Mills, C. B., & Weldon, L. J. (1984). Reading From Computer Screens. (Report No. CAR-TR-94). College Park, MD: The University of Maryland.

Morariu, J. (in press). Designing and evaluating software and software support for the information specialist: A human factors perspective. ASIS Bulletin.

Penna, D. E. (1984). The use of colour graphics display and touch screen to help naive users understand and control a multi-function computer system. Interact 84, 66-70.

Salomon, G. (1977). Interaction of Media, Cognition, and Learning. San Francisco: Jossey-Bass.

Shneiderman, B. (1983). Direct manipulation: A step beyond programming languages. IEEE Computer, 16 (8), 57-69.

Shneiderman, B. (1984). Human Factors Issues of Manuals, Online Help, and Tutorials (Report No. CAR-TR-91). College Park, MD: The University of Maryland.

Shneiderman, B., & Mayer, R. (1979). Syntactic/semantic interactions in programmer behavior: A model and experimental results. International Journal of Computer and Information Sciences. 8 (3), 219-239.

Watkins, M. J., & Tulving, E. (1975). Episodic memory: When recognition fails. Journal of Experimental Psychology General, 104, 5-29.

DESIGNING BLUEPRINT VIDEODISCS FOR FLEXIBLE, MULTILEVEL AND FUTURE USE

Sharyn Neuwirth, MEd; Janis Morariu, PhD

Center for Instructional Development
and Evaluation

University of Maryland University College

ABSTRACT

The multiple-use blueprint videodisc offers a means to reduce production costs of computer-based interactive videodisc instruction. Through a carefully planned combination of computer-generated text and graphics with unfinished frames and audio segments, a single disc can be used to develop several related instructional modules. Flexibility to extend its use to additional audiences and objectives is built-in during the design process. Design considerations for maximal flexibility are discussed.

INTRODUCTION

A powerful new medium for presenting instruction was created by the synergistic merging of computer-based instruction, computer graphics and laser videodisc. The resulting computer-based interactive videodisc (CBIV) instruction offers the capability to overlay computer-generated materials on still or running videodisc sequences combined with the capability to access any frame as needed. CBIV provides a powerful system, capable of efficient delivery of interactive, highly visual instruction, adaptable for numerous levels of users with varied learning needs.

The benefits of interactive video have led a growing number of corporations, organizations and educational institutions to invest in CBIV program development and delivery hardware to meet their educational and training needs.

The typical costs for CBIV development, however, are high. In addition to design and programming costs there is the cost of developing and producing original visual and audio components to be pressed onto the video disc. Production costs (including writers, talent, camera and editing crews) average $1000 per minute of footage, although some of these costs can be defrayed by adapting sections of existing materials. In addition, the cost of mastering a disc can cost up to $3000 per side.

CBIV DEVELOPMENT OPTIONS

Because of the cost, CBIV becomes affordable only when there is a large potential audience or numerous instructional settings to permit economies of scale. Alternative strategies in the development of CBIV instruction need to be explored.

Application-Specific Discs

At one end of the spectrum of CBIV development is the design of audience- and application-specific programs, based on the analysis of a specified set of learners and objectives. The information selected and/or produced for the videodisc is an integral part of the overall planned instruction. The usual assumption with this type of dedicated program development is that the videodisc is designed for a specific CBIV application and is not intended to be adapted for other audiences and instructional settings. Although such a defined and focused approach to instructional development may be the ideal model, the reality of the high cost of production for a relatively limited audience makes such development unaffordable by most educational/training institutions.

Generic Discs

At the other end of the spectrum of CBIV development is the design of generic videodiscs. Both the high cost of producing audience- and application-specific CBIV and the desire to take advantage of the videodisc's potential as a unique information storage-and-retrieval medium, have led to the development of generic videodiscs. As described by Jonassen (1984), a generic videodisc contains a compilation of discrete segments; i.e. still images, text, and audio information on a given topic. The resultant videodisc, then, can be utilized for a variety of purposes. A generic videodisc produced by the National Gallery of Art in Washington, D. C., for example, includes the history of the art museum's development, a guided tour of the gallery, and a catalog of 1,645 paintings, sculptures, drawings and prints in the collection of the museum. This videodisc could be used by librarians, art historians, public school educators, university faculty, and museum exhibit developers to produce either CBIV databases for various artists' work or general genres in art, instructional modules for elementary through university use, or even an information exhibit for museum patron access.

The intent of the generic disc, then, is to include a wide variety of facts and examples on a particular topic/subject, not necessarily considering all the possible audiences and instructional applications. Although a step in the right direction for cutting production costs of CBIV, the ability of the generic disc to satisfy the wide variety of possible instructional needs with appropriate audio and visual information is questionable.

Blueprint Videodiscs

Funding may become available to produce an initial CBIV program for use with a specific audience. The developing institution may already realize, however, that eventually the program is to be expanded to include other audiences or additional training modules as further funding becomes available. Rather than produce an application-specific CBIV program with the limited funds available with the hope of a "new and improved" version at a later time, the developing institution could lay the plans for both the current version and future adaptations in the initial design. A series of planned generic frames and sequences could be added to the videodisc in anticipation of future adaptation or development.

In the realm between application-specific and generic videodiscs, then, is the proposed cost-reducing alternative, the blueprint videodisc. [illegible] blueprint disc is developed not only for the current identified audience, content area and learning objectives, but additional "skeleton frames" or sequences are included for future adaptation or development. The addition of some extra frames or an alternate audio track could conceivably save thousands of dollars and many hours of production and personnel time when additional funding for the continued development becomes available.

Certain frames of the blueprint disc can be designed to allow for specific text and visual instruction to to be supplied by computer text and graphic overlay. Along with computer-generated overlays, the same disc can provide the blueprint and raw materials for several related instructional modules.

Blueprint discs are useful in numerous situations:

-a single blueprint disc can provide the basis for several departments of a large corporation or university to develop specialized variations on related content;

-a disc originally designed to provide initial instruction can be redesigned to provide advanced training;

-a pilot videodisc could be developed and later extended to additional objectives as funding is awarded;

-instruction can be easily redesigned as the curriculum is revised, or updated to reflect innovations in the field.

By providing flexible, redesignable audio and visuals, the blueprint disc defrays the costs of developing related courseware. Additional or future CBIV modules save both video production and disc mastering costs.

Our proposal for developing blueprint discs grew from our experience in redesigning existing discs for the Naval Health Sciences Education and Training Command. Our task has been to redesign existing linear films, which had been pressed on disc, to meet the needs of a new audience and set of curriculum goals. From our modifications of existing frames and segments come our recommendations for designing flexible-use blueprint discs.

DESIGNING A BLUEPRINT DISC

As with application-specific single-use discs, developing a blueprint disc starts with an analysis of learners, objectives and content. In planning a blueprint disc, however, future audiences, constraints, and sets of objectives related to this content are identified to the greatest possible extent. The more specifically the future needs are known in advance, the better the disc can be designed to accommodate them.

Although the initial instructional program from a blueprint disc is designed for a specific audience and set of objectives, the materials included on the disc are designed to accommodate the insertion of alternate sets of material. Computer-generated text and graphics fill in the specific content for current and future lessons. By designing adaptable frames, an alternate set of computer-generated text, labels, and visual pointers can be added later as other instructional materials are designed from the disc. New data can be inserted or updated, and different audio and visual segments of the disc accessed to support new sets of instructional objectives.

Whether producing original materials, adapting existing materials, or pressing a combination of these onto the disc, planning for flexibility guides frame design and selection of materials.

Two principal areas are to be considered:

-designing unfinished visuals

-designing multi-use audio tracks

As with any instructional medium, graphic design and communication principles should be applied. Although issues such as use of color, size, and balance are beyond the scope of this paper, there are several strategies that will make blueprint discs maximally flexible and allow for the best use of computer text and graphic overlay.

Designing Adaptable Visuals

The key to flexible visuals is to press a minimum of written labels or text directly onto the disc frame. In this way varied sets of information can be added later to present other concepts.

A single visual can often support a wide range of topics. For example, an unmarked diagram of the human circulatory system might support text dealing alternately with the heart, the blood vessels, or the direction of blood flow. The audio track and/or computer-generated text and labels add the specific verbal information for each topic. Visual detail is provided by computer-generated graphics, pointers, animation, or highlighting.

The strategy is to design or select visuals which meet the needs of the lesson, but which also allow for the flexible overlay of computer-generated material.

There are six features to consider in designing flexible visuals:

1. Select/design frames that provide sufficient space for the insertion of text. When possible, the main subject should be somewhat in the center of the frame with space available in more than one area.

The same holds true for motion sequences. It is important to allow sufficient space so that labels and titles can be placed over running video if needed. This requires that an area of the frame remain uncluttered during the entire segment.

The placement of open space around the image depends on the amount and kind of text to be inserted. Text requires a large vacant area of horizontal space. A rule-of-thumb is to allow sufficient space to contain approximately 4 or 5 sentences of computer-generated text. A set of labels may require several smaller areas positioned around the frame. Vertical space is best for lists, such as test items with multiple choices, or statements to appear in sequence. The more defined the possible uses for the visual, the more precisely its placement on the screen can be planned.

2. Set aside a fixed area for standard user information, such as computer functions or options. Space must be left clear for the insertion of headings, user options, menu functions and standard operation keys such as page forward or back when a touch screen format is used. It is crucial to carefully plan the size and location of each of these features. Some may need to appear on every frame in a fixed location because they are used consistently by the learner.

When possible, specific kinds of lesson and feedback information should be assigned a standard location on the screen. Fixed placement for user instructions, lesson materials, question items, and feedback helps cue the user to the function of the material. Later, when overlaying the computer text, the use of specific type size, font and color serves to reinforce this association.

In sum, design/select frames that have a minimum of text, labels or visual focusing elements (built in pointers, highlighters, etc.) and are visually uncluttered, with space for the addition of text, labels, graphic overlays and touch-screen options. At the same time, the size of the image and the degree of graphic detail appropriate to the lesson must be considered.

3. An alternative to leaving open areas on each frame is to selectively mask parts of the screen. To cover visual elements that are irrelevant, or to provide a solid background for computer overlays, items can be masked with computer-generated blocks of any size and color. To simplify masking, items that may be relevant only in certain lessons should be placed where they can be easily isolated and covered. However, it is usually preferrable to add detail, labels or pointers than to delete details with a mask overlay that may further clutter the screen.

4. Consider the type of graphics to include. After analyzing current and possible future needs, the type of materials needed to match instructional objectives should be specified. Are close- ups needed? magnification? live action? first person demonstration?

For example, to demonstrate blood flow, an action sequence may be more appropriate than a still frame, and a live action view of a real heart and blood vessels may be more or less useful than an animation, depending on the objectives of the lesson.

The content and focus of each segment, and the type of visuals needed are determined by both the current and anticipated learning tasks and objectives.

5. Plan the introduction of images so that they are meaningful when shown in other sequences. For example, in planning the current visuals, identify items which might not be easily recognized in a different context. Extreme close-ups of a cell nucleus will be recognized within a lesson on cells. However, if these frames are used later in a module on structural tissue, the meaning of the visual may not be clear. If such needs are anticipated, frames which orient the learner, such as long shots and zooms to the subject, can be included.

6. In designing for maximal flexibility, allow for several shots of a subject, at varied distances, perspectives, and areas of the screen. If the action is slow enough, several of the individual frames which make up a zoom or an action sequence can be used as stills to support printed text. When action sequences involve any fast movement, however, these frames appear jittery in playback and are not usable as stills. In this case, several still frames that show progressive stages in the action need to be included. Such stills are invaluable as background images for computer text when instruction requires a step-by-step explanation of a process or procedure.

In summary, select/design a variety of stills and action sequences. Include a large number of still frames that can support a range of printed information. Include a variety of action sequences which present the key item from several perspectives and in different parts of the screen.

Designing Multi-Use Audio Tracks

The next major consideration is the audio track. Although there are two tracks per disc side, each is limited to 1/2 hour running time. However, the development of accessory audio peripheral devices may soon allow blueprint discs greater flexibility in presenting alternate material. Meanwhile, the two-track limitation requires careful planning of the audio.

Three considerations apply:

1. Plan for re-usable phrases and explanations within the running narrative. Consider using one track for the audio to support the initial module to be developed. In scripting this, however, it is possible to keep an eye to future use by conceiving of each sentence, explanation or definition, as a potentially independent unit. Writing simply, towards an average-level user, and providing appropriate pauses in the narration allows such segments to be accessed independently. Words, phrases and key explanations that may be needed later should be recorded on the disc so that they accompany relevant visuals. In this way, associated audio and visual segments can be accessed separately, or together.

2. Consider designing the second track to store supplementary audio segments for future use.

The second audio track can store sound effects, music, quotations, vocabulary items, etc. Placing these on the disc so that they relate to the visuals offers the designer the option of providing playback with either running video or with any number of computer-generated graphics. The sound effects on one audio track can also be played simultaneously with the narration on the other audio track.

If the future use is known, and the audience and objectives identified to some extent, the second audio track can be used to carry particular content for parts of the projected lesson. It can hold key vocabulary terms for creating a pronunciation guide and glossary.

The second audio track can be used to present alterntive explanations of the accompanying video for other users. For example, if the current project is to train beginning emergency medical technicians, the explanations on the first audio track may be brief and simple. If the projected use is experienced technicians, the second audio track could offer more advanced, detailed explanations for the same visuals.

The second track may also carry portions of the lesson, such as the overview or summary. The more defined your future audience or objectives, the more specialized the audio can be.

3. As in designing the visuals, designing the audio also involves creating adaptable verbal sequences, with the remainder to be filled with computer-generated text. Which verbal information to press onto the audio track and which to provide via computer text must be decided. Three questions can guide each decision:

-Which information may need to be updated later to reflect current data or knowledge in the field?
Material presented as computer text is easily revised. There is no way (at present) to revise spoken material on the disc.

-Which information will be better understood if read, or heard?
If learning pronunciation is important, the audio track should present the material. To allow the learner some control of the pace, repeat functions can be offered. On the other hand, printed text allows the learner to see the word, and to pace the rate of presentation.

-Which information may be needed in other modules built from this disc?
Present information that is specific to a single module through computer text. Save the audio track for reusable verbal sequences.

In sum, plan the first audio track for the current use, to closely accompany the running video. Use the second track for sound effects and explanations designed for an alternate audience. Assign to the computer material that does not require audio; for example, information more easily presented with a still image, and information that will later be updated or revised.

The Blueprint Index

The final step in the production of a blueprint disc is to provide an audio/visual index that will assist in future adaptations. This index is a catalog of each discrete audio and visual segment, along with its exact beginning and ending frame numbers. This provides the designer with a list of the raw materials that can be modified or re-combined into new instructional sequences.

A summary of decisions made in designing the disc will also aid in building new modules. Conventions for the use of specific colors, areas of the frame assigned to certain kinds of text, and range of user options specify some of the features of the disc that can be exploited in its redesign.

CONCLUSION

Planning for future needs while designing for current ones can result in the significant reduction of design and production costs. The procedures and guidelines outlined in this paper are intended to facilitate the development of a blueprint videodisc and accompanying computer support that can be extended to new audiences and expanded to new sets of objectives. The blueprint disc provides the framework and the raw materials from which other structures can be created.

REFERENCE

Jonassen, David H., "The Generic Disc: Realizing the Potential of Adaptive, Interactive Videodiscs." Educational Technology, January, 1984.

THE CONTEXT OF A CAI LESSON

J.D. Bradley, L. Glasser* and S.W. Orchard

Department of Chemistry
University of the Witwatersrand, Johannesburg
2001, South Africa

ABSTRACT

Recent experimental work by ourselves has demonstrated distinct student preference for a tutorial mode of instruction via PLATO rather than a simpler practice and guidance mode offered on a microcomputer network. We hypothesize that weak students use the tutorial mode to replace self study. We report on an experiment to examine the effectiveness of self study in supplanting tutorial support, and demonstrate that student perceptions of CAI lesson quality are markedly affected by their prior processing of the lesson material. Interaction between learning activities must be taken into account in studies of effectiveness.

We recently reported (Glasser and Orchard, 1985) on a novel comparison of student acceptance of CAI instruction on a sophisticated main-frame system, PLATO, as against CAI on a microcomputer-based local area network (viz., 15 Apple II microcomputers linked to a central 10Mb hard disk via a ProNET local-area network). An interest in the competitiveness of microcomputer CAI is appropriate in view of Ploeger's (1985) observation of a developing tendency for improvement in the style of microcomputer CAI as such systems become more generally available.

The experimental class consisted of about 60 medical-auxiliary students with a weak University-entrance record, divided into two roughly-matched groups. These groups were offered CAI practice, at a rate of one 50-minute period a week, on three topics, viz., Chemistry Introduction, Thermodynamics, and Chemical Equilibrium. Each topic was covered in 4 to 6 weeks, alternately on PLATO and on microcomputer for each group.

The program material was selected to provide lessons of comparable standard on each system, matching the demands of the lecture course in which context the supplement CAI was provided. There are, however, notable differences in style (Alessi and Trollip, 1985) between the PLATO and the microcomputer lessons. The PLATO lessons generally take much longer (typically 45 minutes, as opposed to 20 minues for the Apple), and are in a more-sophisticated, tutorial mode. The microcomputer lessons are of a practice and guidance type, consisting of a brief statement of onjectives followed by an interactive "question and answer" session (Bradley, et al., 1984).

The students completed three anonymous questionnaires at intervals during the 14 weeks of the experiment, which was carefully monitored by the staff member in charge (SWO). A distinct preference was expressed for the PLATO system, which preference increased with exposure to the systems. This preference has been ascribed by us, in part, to the tutorial style of the PLATO system (much appreciated by these academically weak students). Other system features which contributed to the favourable assessment were the HELP facility, the very flexible high-resolution graphics, the page-size touch-sensitive screen, and a number of special function keys.

The environment also contributed to the preference: the quiet, carpeted, uncrowded conditions of the PLATO terminal room contrasted very favourably with the noisy and uncomfortable circumstances of the Apple network room.

On the other hand, the microcomputer system is considerably cheaper, and its much simpler operating system made it attractive to the staff member involved in program selection and student monitoring. It proved to be generally the more reliable system since it was independent of telephone lines for communication. This represented a marked advantage over the unreliability of communications on the PLATO installation at our disposal.

The Context of the Lesson

Given the evidence of some significant advantages of the microcomputer network, in the face of the technological strength of PLATO, it seemed worthwhile to explore, in greater depth, which of the disadvantages of the microcomputer are inherent in its less-advanced technology and poorer resources of memory, and which are simply facets of the different styles of presentation and of mode of usage by our students.

It thus seemed important to examine the circumstances of the CAI lesson rather more carefully, while attempting to control some of the variables that were not controlled in the earlier study. For example, our impression is that many students in that study used the CAI lessons "cold", that is, either with no preparation (or other than memories of school-work) or only with the preparation that is provided by a prior lecture. Indeed, it is common experience that weaker students do not follow-up on their lectures, do not read beyond their (often poor) lecture notes, and rely on "cramming" to pass examinations. For such students, tutorial-style CAI lessons (as on PLATO) are likely to be perceived as more valuable than our practice-and-guidance-style microcomputer CAI lessons. The latter, in order to be used effectively, require some intervening study and processing by the student of the material introduced in the lectures. In support of this contention we note that the microcomputer lessons, in the absence of PLATO for comparison, have been viewed very favourably by underprepared science students, where the lessons have been specifically integrated into a highly structured and specially-designed curriculum (Bradley, et al., 1985).

Thus, we wished to explore how student perceptions of the value of our practice-and-guidance CAI lessons are influenced by the extent of their prior processing of the lesson topic. This needed to be studied independently of the differing sophistications and working conditions of our microcomputer versus PLATO experiment, and without the confusion of the different operating systems.

We have selected a similar group of students to those in our original experiment and have induced them to take a series of our microcomputer lessons, at various stages in their instructional programme. At their own choice, they have taken the microcomputer lessons at all stages during the other course activities, i.e., from CAI prior to any other instruction through to CAI subsequent to all other instruction. The CAI lessons used were, predominantly, the same as those used by students in our earlier study (Glasser and Orchard, 1985), although improvements had been made to some of them.

Experimental Design

Seventy science students registered for the specially-designed curriculum already mentioned (Bradley, et al, 1985) were asked to complete short, anonymous questionnaires (see Appendix) immediately on finishing each CAI lesson. The questionnaires asked whether or not they had undertaken previous study of the lesson topic through each of lectures, small group tutorials, laboratory experiments, reading of the textbook, or attempting written assignments. In addition, the questionnaires asked for opinions (on a 5-point scale) concerning the clarity of purpose (objective) of the lesson, the clarity of presentation, the usefulness of the hints, the ease/difficulty of the lesson and, finally, the knowledge gained in the CAI lesson concerned.

Each of these five questions had appeared in the larger questionnaire administered in the earlier comparative study of Glasser and Orchard (1985).

The experiment ran for about 14 weeks, during which period these students could have completed about 1400 lessons - and hence 1400 questionnaires - altogether. Completion of the questionnaires was voluntary, however, and in fact only 402 (about 30% of the maximum possible) were completed.

In addition to the individual CAI lesson questionnaires described above, all 70 students completed a retrospective questionnaire on the whole year's CAI lessons, which included the same five questions, together with other questions about their mode of use of the CAI lessons and the perceived value of these lessons as compared to other learning activities in the students' curriculum. Eleven of the students completing the retrospective questionnaire were further interviewed in private to elucidate details of the origins of their views. However, for the purpose of this paper, the analysis of the responses to the individual lesson questionnaires is of the greatest interest, and our discussion therefore focusses on this. Indeed, the restrospective questionnaires did not reveal any substantially differing information.

Analysis

The collection of responses generated was coded for computer analysis, and was analyzed by means of the SAS statistical analysis system on our IBM 3083 mainframe computer. Various tabulations of the question responses against one another, as well as combinations of responses, were generated and examined. The statistical measures yielded by SAS provide qualitative support for the outcomes and intepretations described below.

In order to obtain an objective comparison, we converted the distribution of responses to each question to percentages (to provide a common base), and a total "score" for the questions was calculated by assigning a weighting of 2 for the most "favourable" response (i.e., very clear, useful, easy, etc) through zero for a neutral response, to -2 for the most "unfavourable". This yields a possible range of +200 to -200 for the score on the question. It may be argued that this linear progression does not correctly represent the students' strength of response, but it seems to us to be as good as any other, also arbitrary, choice of progression.

Outcomes

In support of previous experience (Bradley, et al., 1985, see also Lamprecht and Pretorius, 1984), student response to our collection of CAI material and its operation was rather positive;

the overall "scores" for questions 1 to 5 were 104, 118, 61, 44 and 57, respectively. Indeed, in the final open-ended question in the retrospective questionnaire on "Useful Features" of the CAI, many spontaneously answered "all" and, correspondingly, answered "none" to "Irritating Features". This was not uniformly the case, of course, and some resistance was expressed to the inflexibility of the system (e.g., insistence on correctness of spelling, and inconsistent range allowable on numeric responses), as well as to the unreliability of recording of marks. We conclude from this portion of our analysis that our microcomputer system of CAI provides a learning experience which is perceived by students to be satisfactory and worth the expenditure of time and effort. This is confirmed by the responses to the question in the retrospective analysis as to the relative preference for microcomputer work as against small group tutorial work; many more expressed preference for microcomputer work than for group work, though support for the balanced combination of the two generally prevailed ("score" = 36).

The outcomes with respect to the interactions of the various modes of learning upon one another have been most interesting and illuminating. Even a casual inspection of the responses shows, quite clearly, the markedly better rating of a CAI lesson following almost any previous study of the lesson topic by other means. The weakest influence was generally exerted by previous laboratory activity, an outcome that is consistent with direct questions about the perceived value of the laboratory work (Bradley, et al., 1985, and retrospective questionnaire). The strongest influences were generally exerted by previous lectures and tutorials, with the greatest impact of these being upon the students' perception of knowledge gained. In the latter case, prior treatment of the topic in a lecture led to a 120% improvement (relative to the absence of a prior lecture), whilst prior tutorial discussion led to a 82% improvement in perceived knowledge gain. The clarity with which the purpose or objective of the CAI lessons was perceived was also markedly improved by a prior lecture (23%) and tutorial (24%). Even the clarity of presentation was perceived to be improved by prior familiarisation with the topic provided by a lecture (29%) or tutorial (21%).

Conclusions

The outcomes described demonstrate the positive impact of prior learning activities on students' perceptions of our practice-and-guidance microcomputer lessons. This supports, but does not prove, our hypothesis that part of the observed difference in students' judgements of tutorial-style lessons (on PLATO) and practice-and guidance-style lessons (on microcomputers) is attributable to the differing impacts of a lack in prior learning activities. However, our present results suggest that, in the absence of human tutoring, those lessons that provide a computer-based equivalent will be perceived more favourably. Since the present experience demonstrates that our students often tackle the CAI lessons with little pre-processing of the lesson topic, the tutorial crutch will be a vital component of the lesson. Indeed, the improvements in perceived knowledge gain (98%) and other factors when comparing tutorial-style with practice-and-guidance style lessons (Glasser and Orchard, 1985), are comparable in magnitude to the improvements resulting from small group tutorials reported above. This conclusion should, by no means, be taken as advocacy for self-contained tutorial lessons as a substitute for human intervention. A recent report on PLATO usage in a cross-cultural study (Laubli, Pope and Hinton, 1985) "showed that careful supervision of a trainee's interaction with such lessons was needed [to reduce] boredom, frustration and confusion with the structure of [the] lessons".

The implications of these outcomes have a wider significance, however. Firstly, they demonstrate how strong can be the interaction between learning activities, and how very favourable to learning this interaction can be when the activities are consonant rather than dissonant. Secondly, in relation to the numerous studies of the effectiveness of CAI (Kulik and Bangert-Drowns, 1983; Bangert-Drowns, Kulik and Kulik, 1984), it is evident that considerable variation in effectiveness may be found in different studies, depending on the nature and strength of this interaction and, indeed, on the quality and suitability of the CAI material for the purpose in hand (which remains too often unquestioned).

Finally, we should draw attention to the fact that these conclusions pertain to students' perceptions of the quality of our CAI lessons. We have not attempted to measure the effectiveness of the CAI, or to provide other measures of its quality. Indeed, it is our philosophy that there is no independent measure of one learning activity in the absence of others; we rather take the holistic view that the totality of the learning experience is greater than the sum of its parts (Boeyens, 1985).

Literature Cited

Alessi, S.M., and Trollip, S.R. (1985) "Computer-Based Instruction: Methods and Development", Prentice-Hall, Inc., N.J.

Bangert-Drowns, R.L., Kulik, J.A., and Kulik, C-L.C. (1984) Proc. 25th Intl. ADCIS Conf., Columbus, OH, pp 168-171.

Boeyens, J.C.A. (1985) inaugural lecture, University of the Witwatersrand, Johannesburg.

Bradley, J.D., Brink, G., and Glasser, L. (1984), "Project Report: WITSCHEM", 25th Intl. ADCIS Conf., Columbus, OH p.226.

Bradley, J.D., Brand, M. and Brink, G. (1985) S. Afr. J. Educ., 5, [illegible]-50.

Glasser, L., and Orchard, S.W. (1985) "Micro and Mainframe CAI: A Study in Contrasts", submitted for publication.

Kulik, J.A., and Bangert-Drowns, R.L., (1983) J. Educ. Technol. Systems 12, 137-158.

Lamprecht, G.J., and Pretorius, E.V.E. (1984) ChemSA 10, 292-3.

Laubli, M., Pope, M., and Hinton, T. (1985) Programmed Learning and Educational Technology (PLET), 22, 224-229.

Ploeger, F.D. (1985) The Effectiveness of Microcomputers in Education. Doc. ED 246 876, ERIC Clearinghouse on Information Resources, Syracuse University, Syracuse, NY.

APPENDIX

UNIVERSITY OF THE WITWATERSRAND, JOHANNESBURG
DEPARTMENT OF CHEMISTRY

CAI PROGRAM SURVEY

PROGRAM NO: ________________ DISK: ________________ DATE: ________________

COMPLETION TIME: __________ Minutes

Before using this program you may have received instruction or done work in tutorials or on your own on the subject matter of the program. Please indicate with a tick (✓) your prior study in this area below.

	HAD LECTURE	HAD TUTORIAL	HAD LAB. EXPT.	READ TEXTBOOK	WORKED QUESTIONS
YES					
NO					

The following questions should all be answered by an integer number on a scale of 1 to 5, which most closely matches your feelings (3 generally represents fairly neutral feelings on the matter). Mark your rating with a cross e.g. X.

1. How clear was the purpose (objective) of the program?

   1 2 3 4 5
   very clear — very unclear

2. How clear was the presentation of the program? (e.g. were screen displays clear, was it clear what was expected of you, etc?)

   1 2 3 4 5
   very clear — very unclear

3. How useful were the hints or the assistance from the computer, when you were unable to provide correct answers in this program?

   1 2 3 4 5
   very useful — no use at all

4. On average, did you find this program?

   1 2 3 4 5
   much too difficult — much too easy

5. How much has this program helped you to improve your knowledge in the relevant area of chemistry?

   1 2 3 4 5
   very much — not at all

BEST COPY AVAILABLE

Teaching Cognitive Strategies to Native Students through Integrating Instrumental Enrichment and LOGO Programming

W. Bruce Clark
Institute for Computer Assisted Learning
The University of Calgary
Calgary, Alberta, Canada

Kathleen Wenger
Department of Curriculum and Instruction
The University of Calgary
Calgary, Alberta, Canada

Abstract

This paper presents a preliminary report on a project which tested the feasibility of aiding Native junior high school students to improve their cognitive skills through LOGO programming combined with techniques borrowed from Instrumental Enrichment. LOGO was selected in the hope that graphics programming would provide a cultural bridge and that using microcomputers would in itself be positively motivating. From instrumental enrichment was borrowed the classification scheme of cognitive functions, the four-step lesson plan, and the tests used as measures of cognitive functioning. Twenty-five lesson guides were prepared and three workshops were developed for participating teachers. A pilot project was implemented using as subjects 71 Native Cree students in four classes. At the end of the first phase of the project anecdotal evidence for the effectiveness of the approach was found, but virtually no gains were found in the empirical tests. In the second phase, in addition to LOGO, parallel programs were implemented in mathematics, reading and writing, and emphasis was placed upon the transfer of cognitive skills from one area to another. At the end of the second phase positive anecdotal results were supported by gains in the empirical measures.

Overview

This paper reports on the first phase of a project which tested the feasibility of aiding Canadian Native junior high school students to improve their cognitive skills through LOGO programming combined with techniques borrowed from Instrumental Enrichment (Feuerstein, 1980). It will describe briefly the general nature of the problem, the rationale for selecting an integration of these two techniques, the nature of the program developed, and a preliminary analysis of the results. It should be noted that the program was implemented in two phases, with the second phase including additional elements directly addressing mathematics, reading, and writing. Since the data for the second phase have not yet been analyzed, this paper will restrict itself to Phase 1 which concentrated upon using LOGO. (The results from Phase 2 can be reported at the conference.)

Background

Over the past several decades many attempts have been made to provide special programs to improve the learning achievement of Native adolescents. Such programs have usually been based on one of two philosophical approaches: (1) the remedial model which concentrates on the development of basic concepts and terms used in the educational process, on improving verbal ability, on upgrading academic skills, and on developing manual skills; or (2) the supplementary model which emphasizes the development of self-concept and cultural identity. Given the recurring high rate of dropout and age-grade retardation found among Native adolescents (Committee on Tolerance and Understanding, 1984), both approaches have been of limited effectiveness. In contrast to the reality of much of Native education, several investigators have suggested that properly planned formal schooling should provide special kinds of learning experiences (Greenfield and Bruner, 1969: Neisser, 1976; Scribner and Cole 1973). Theoretically such experiences should provide opportunities for students' thoughts to be freed from the immediate context. They should be enabled to consider problems hypothetically, to imagine alternative solutions and situations, and to evaluate these with no concern for everyday practicalities. Vigotsky (1978), in discussing the influence of formal school learning on the cognitive development of the child, contends that, while learning is not the same thing as development, properly organized learning does result in cognitive development. Native adolescents who may not have had access to such properly organized learning are at a disadvantage.

Instrumental Enrichment.

One approach to aiding the development of cognitive functions is that development by Reuven Feuerstein (1980) known as Instrumental Enrichment (IE). Feuerstein contends that the development of cognitive functioning is the result of two influences: (1) direct exposure to stimuli, and (2) the transformation of such stimuli by some mediating agent, usually a parent, sibling, or other caregiver. The mediating agent selects and organizes the world of stimuli for the child. Such mediation is intended to transcend the immediate needs or concerns of the recipient of the mediation by venturing beyond the here and now in space and time. For example a mother may use the spoon with which she has finished feeding the child and demonstrate for the child other things which can be done with spoons,such as tapping them on objects of different density and producing different sounds. With older children one begins to provide reasons for one's requests so that the child may participate in the reasoning associated with such requests.

Feuerstein, using an input-elaboration-output paradigm, has identified a series of potential deficient cognitive functions. At the input phase some possible impairments include:

- blurred and sweeping perception;
- unplanned, impulsive, and unsystematic exploratory behaviour;
- lack of, or impaired, receptive verbal tools and concepts which affect discrimination;
- lack of, or impaired, spatial orientation, including the lack of stable systems of reference which impair the organization of space;
- lack of or impaired, capacity for considering two sources of information at once.

At the elaborational phase deficiencies include:

- inadequacy in experiencing the existence of an actual problem and subsequently defining it;
- inability to select relevant, as opposed to irrelevant, cues in defining a problem;
- lack of, or restricted, inferential hypothetical thinking;
- lack of, or impaired, strategies for hypothesis testing;
- lack of, or impaired, planning behaviour.

Impaired cognitive functions affecting the output phase include:

- trial-and-error responses;
- lack of, or impaired, need for precision and accuracy in communicating one's responses;
- deficiency of visual transport.

(For the complete list of cognitive deficiencies see Feuerstein, 1980,73-75).

Instrumental Enrichment is a remedial program developed by Feuerstein to address such cognitive deficiencies. This program makes use of specially designed tests to identify cognitive deficiencies, and a special set of content-free student materials designed to provide an environment through which students may further develop their cognitive functions. The program is implemented by teachers who have received special training in the methods of focusing attention on those functions that are required for the mastery of each task. Also it falls to the teacher to orient the child toward bridging from IE exercises to broader behaviour in school and everyday life. Each IE lesson includes four stages: (1) introductory discussion, (2) independent work, (3) discussion and the development of insight, and (4) a brief summary .

Despite the attractiveness of Instrumental Enrichment and its many success stories, it has not proved particularly successful with Native adolescents. One possible explanation is the dependence upon paper and pencil exercises which appear to have a demotivating effect upon this population. For that reason the researchers involved in this project decided it was necessary to look beyond IE in its pure form as a device for assisting the development of cognitive functioning.

LOGO Programming

A second possibility which suggested itself was the use of the computer programming language, LOGO, as its use is described by Papert (1980). In developing LOGO, Papert and his colleagues set out to create what they describe as an environment for learning. LOGO takes advantage of intrinsic geometry allowing the turtle to be moved on the computer screen with reference to its current location as opposed to a fixed set of coordinates. This in turn means that there can be a correspondence between the way in which a child can physically move about a room and the directions which can be used to move the turtle to achieve a comparable movement (e.g., FORWARD, RIGHT TURN, BACK). For this correspondence Papert coined the expression, body syntonic. Another valuable feature of LOGO is that it easily supports learning structured computer programming--new procedures can easily be constructed based upon previously constructed procedures.

One can notice a similarity between some of the claims made for LOGO and those made for Instrumental Enrichment. LOGO lends itself to supporting the development of such learning strategies as (1) breaking problems into manageable sub-problems, (2) mean-ends analysis, (3) debugging programs which do not perform as desired, (4) positive attitude toward errors. Many claims have been made for the generalizability of these skills from the LOGO environment to general problem-solving skills. The research on such generalizability has been reviewed by Krasnor and Mitterer (1984). It is their observation that most of the evidence in support of generalizability is anecdotal in nature, and that there is virtually no empirical evidence that the powerful ideas claimed by Papert in fact generalize to other domains. They further conclude that educators interested in using LOGO should tailor the LOGO environment to facilitate transfer, including the use of techniques which foster awareness of the general utility of problem-solving heuristics.

Native Students and Computers

It had been the experience of the researchers in a previous project at Old Sun College in Gleichen, Alberta, that microcomputers are not only an acceptable form of technology for Native students, but that they appear to have considerable positive motivational influence, at least in the short term. The experience at Old Sun College was with adults rather than adolescents, but that program did include some programming in LOGO. It was observed that the students in that course (a) did not appear to have any cultural bias against the use of computers and (b) tended to use LOGO to construct patterns similar to their Native art. This tendency to experiment with graphics patterns akin to Native art suggested to the researchers that LOGO might be used to provide a cultural bridge through a relationship to Native art.

The Experimental Program

After weighing the strengths and weaknesses of both Instrumental Enrichment and LOGO, it was decided that some combination of the two might reasonably be expected to aid in the development of cognitive functioning of Native junior high school students. It was anticipated that LOGO would provide a cultural bridge and that using microcomputers would be positively motivating. From Instrumental Enrichment was borrowed the classification scheme of cognitive functions, the four-step lesson plan, and the tests used by Feuerstein as measures of cognitive functioning (Learning Potential Assessment Device, and Representational Stencil Design test). An analysis of possible LOGO activities indicated that most (though not all) of the cognitive functions could be addressed. While most exercises call upon more than one cognitive function (as do most of the exercises in Instrumental Enrichment), it was judged possible to focus student attention upon selected functions.

One major difference between the approach selected and Instrumental Enrichment (and one which tended to haunt the project) was the fact that LOGO is not content-less in the way that the IE materials are. Once can learn to program, and a temptation which had to be resisted by the teachers participating the project was to turn the program into a course on programming rather than concentrating upon cognitive function development.

Krasnor and Mitterer's concern over the absence of empirical evidence to support the claims for transfer of skills from the LOGO environment to other environments was also taken seriously. It was decided that the cooperating teachers would be asked to maintain attendance records and logs during the process to provide an anecdotal record. However, scores on the Learning Potential Assessment Device and on the Representational Stencil Design Test were also adopted as dependent measures of the effectiveness of the intervention. It was further decided to test a group of students in a comparable school at the same times as the intervention group was tested. This second group could not be legitimately referred to as a control group given all the extraneous influences that could be present and the fact that it was considerably smaller than the treatment group, but it would serve as a partial check for maturation and history.

Materials Development & Teacher Training

Since the planned program involved having the students learn LOGO, it was necessary that the sequencing of the content be such that it would facilitate that endeavor as well as address specific cognitive functions. In developing the materials, the project first developed an outline of how students would be introduced to LOGO, starting with an introduction to the computer, then an introduction to LOGO, LOGO primitives, simple procedures, conditionals, and so on. On a second pass, each activity was analyzed for the cognitive functions which would be called upon and the functions which would be targeted for attention were identified. It was decided that every activity in the program would be treated as an occasion for instrumental enrichment, including becoming acquainted with the computer. The following are examples of cognitive functions singled out for attention for selected activities:

Activity	Cognitive Function
Introduction to parts of computer	Adequate labelling
Turning on computer and booting operating system	Adequate labelling Appreciate inter-relatedness of events Temporal orientation
Introduction to programming	Precision and accuracy in communication
LOGO primitives Adequate labelling	Spatial orientation
"Playing turtle" Making of comparisons Troubleshooting errors	Perceiving and defining problems

Twenty-five lesson guides were prepared for the teachers which identified the activity, its primary and secondary goals, subgoals, suggestions for motivation, suggestions for presenting the class activity, for student participation, and for the discussion for insight. Suggested graphics designs which could be used were included, along with the appropriate program code for the teacher. No student print materials were prepared on the advice that such materials would be rejected by the students. (This advice turned out to be inappropriate, and students requested handouts.)

Three workshops were developed introducing teachers both to LOGO and to the techniques through which it was to be taught in order to facilitate the growth of cognitive functions.

(Additional materials in the areas of mathematics, reading, and writing and associated workshops were prepared for the second phase of the program but are not being reported here.)

Pilot Test

Subjects During January and February, 1985, a pilot test was conducted using as subjects 71 Native Cree students at schools in Hobbema, Ponoka and Wetaskiwin, Alberta. All had been previously identifies by school administrators and teachers as underachieving students; i.e., with age-grade retardation of one or more years. Students were grouped in six classes as part of their on-going school program. These included the special class at Ponoka Junior High School (8 students), classes 7B and 8B at Ermineskin Junior High School, Hobbema (19 and 11 students respectively), a class under the auspices of the Ermineskin Achievement Achievement Training Project (EATP) at Hobbema (21 students) and grades 7 and 8 at Queen Elizabeth Junior High, Wetaskiwin (12 students). It should be noted that the EATP class in Hobbema was somewhat distinct in that its students were attending this alternative school program voluntarily, all having previously dropped out of the regular school system for a variety of reasons. The subject group comprised 34 females and 37 males whose ages ranged from 12 years to 17 years (mean = 14.9 years).

Due to restrictions of geography and school administration it was necessary to use intact groups. Thus it was not possible to randomly assign students to groups or to match groups on the basis of preliminary test results. The grade 7 and 8 classes at Queen Elizabeth Junior High School in Wetaskiwin were designated as the non-intervention group, while the remaining classes comprised the intervention group.

Four teachers participated in the project, none of whom had prior experience with LOGO, and only one of whom had had previous experience with instrumental enrichment.

Teacher Workshop Two workshops for the teachers were conducted prior to beginning the program, and a third was held at the end of the second week. At least one member of the research team visited the project every week to observe what was happening and to provide assistance for the teachers. Teachers were requested to maintain logs which recorded traces of progress (and setbacks), and difficulties they encountered .

Computer Labs A computer lab consisting of six Apple //e microcomputers was provided by the Ponoka School District for the 12 students in Ponoka Junior High School. The Canadian Centre for Learning Systems in Calgary supplied a lab of 10 Apple //c microcomputers for use by the classes in Hobbema. The lab was located in the facilities of the Ermineskin Achievement Training Project and classes from the Junior High School next door came across to use the lab.

Time on Task Because of scheduling circumstances within the schools involved, students in Ponoka Junior High School and Ermineskin Achievement Training Project had 1 1/2 hours per day during regular school days in the program. Students from Ermineskin Junior High School averaged approximately one hour per regular school day.

Testing Prior to beginning the program students in both the intervention and non-intervention groups were pre-tested using a battery of tests consisting of Raven's Colored Progressive Matrices, Sets C, D, and E; and the Representational Stencil Design Test, Set 1. After the 6-week intervention program, students were again tested, this time using the Learning Potential Assessment Device, Variations I and II; and the Representational Stencil Design Test, Parallel Form. Identical pre- and posttests were not used in order to avoid problems associated with the repetition of identical measures. It was erroneously assumed that the Raven's Colored Progressive Matrices sets used and the Learning Potential Assessment Device (LPAD) were equivalent tests. In fact, the LPAD is an equivalent test to a subject of Raven's. This was taken into account in analyzing the data.

For convenience later the pre-post test pair consisting of the Raven's A, Ab, and B and Learning Potential Assessment Device I will be referred to as LPAD1. That consisting of Raven's C, D, and E and Learning Potential Assessment Device II will be referred to as LPAD2.

Data Analysis Two sets of data were available for analysis--the anecdotal data available from the teachers' logs and the researcher's observations, and the statistical data from the pre- and posttests. Interestingly, if one were to base one's judgment upon the anecdotal data, one would conclude that the project was successful. The statistical data, on the other hand, tell a different tale.

The observations of teachers and researchers indicate a change in behavior of the students. The motivational effects predicted for the program were confirmed. Attendance records for the intervention groups were higher than for the non-intervention groups. Students in one class which were notorious for communicating by shouting and throwing things adopted more appropriate behavior. Even some parents of students in Ponoka confided in the researchers that they noticed a difference in their children at home, particularly with respect to their willingness to attend school. The logs document improvement in communication skills, increased systematic exploration, improved temporal and spatial orientation, improved inferential thinking, improved ability to perceive and define problems, reduced impulsive behavior. Many students not only requested handouts, but on their own initiative developed systems for saving their work.

Analysis of the test results, however, are not nearly as encouraging. To compensate for differences in the number of items in the pairs of tests, Feurstein's method of converting raw scores to percentages was followed. No statistically significant effects due to gender or age were found. A 2 (treatment) x 2 (occasion) mixed multivariate analysis of variance was performed on the three dependent variables represented by LPAD1, LPAD2 and Representational Stencil Design, with Treatment (intervention vs non-intervention) as a between-subjects variable and Occasion pretest vs posttest) as a within-subjects variable. These were followed by univariate analyses, using the same design, for each dependent variable taken separately. Analyses were carried out using the general purpose analysis of variance and covariance program, BMDP4V.

The 2 x 2 mixed multivariate analysis of variance revealed a significant effect of Occasion on the combined dependent variables ($F(3,40)=22.96$, p, $<.001$>. No significant effects due to Treatment ($F(3,40)=1.8$, $p >.05$) or the interaction of Treatment with Occasion ($F(3,40)=2.75$, $p >.05$) were found. Univariate analyses, with the same design, for each of the dependent variables taken separately showed significant effects of Occasion on LPAD1 and LPAD2 ($F(1,42)=32.89$, $p < .001$) and $F(1,42)=33.40$, $p < .001$) respectively. HOWEVER, this effect was due to lower posttest mean scores on both variables.

Only one other univariate finding was significant--for the effect of the Occasion by Treatment interaction on Stencil Design scores ($F(1,42)=4.75, p< .05$). Simple main effects tests demonstrated that this effect can be attributed to the fact that posttest mean scores were significantly higher for the Intervention group ($F(1,42)=21.25$, $p >.05$), but not for the Non-intervention group ($F(1,42)=0.14$, $p > .05$). At most this hints at a possible positive treatment effect on those aspects of cognitive functioning measured by the Representation Stencil Design Test.

Discussion

From the foregoing one concludes that, though there was some anecdotal evidence to suggest that this program in some measure achieved its objectives, the test scores do not bear out this conclusion. The pessimistic view would be that this approach combining features of Instrumental Enrichment with LOGO programming is ill-conceived. Before one yields to this conclusion, however, there are a number of alternate explanations which are being pursued.

In the first place, there is now serious doubt that the pre-and posttests were indeed equivalent forms, as was originally believed. This in turn casts doubt upon the validity of the empirical measures. Even if the measurement devices are valid, the 6-week time period may well have been too short to have any significant effect. Feuerstein, for a different audience, recommends two to three years. This will be examined in the analysis of test data from the second phase of the project. A third explanation, in part associated with the short time frame, is that the need to learn LOGO became a dominant concern and insufficient attention was paid to Instrumental Enrichment during this first phase of the project. This would suggest a need to re-examine the initial lesson materials, and also to refine the training program for participating teachers.

In short, there is reluctance to write off this approach to dealing with this student population until all the data from the project have been analyzed, and satisfactory explanations for the discrepancies between the anecdotal and empirical data satisfactorily explained.

Bibliography

Committee on Tolerance and Understanding. (1984). Native Education in Alberta: Discussion paper No. 2. Calgary, Alberta.

Feurstein, R. (1980). Instrumental Enrichment: An intervention program for cognitive modifiability. Baltimore: University Park Press.

Greenfield, P.M. and Bruner, J.S. (1969) Culture and cognitive growth. In D.A. Goslin (Ed.), Handbook of socialization and research. Chicago: Rand McNalley and Co.

Krasnor, L.R. & Mitterer, J.O. (1984). LOGO and the development of general problem-solving skills. Alberta Journal of Education Research, 30:133-145.

Neisser, U. (1976). General, academic, and artificial intelligence. In L. Resnick (Ed.), The nature of intelligence. New York: Erlbaum.

Papert, S. (1980). Mindstorms: Children, computers and powerful ideas. New York: Basic.

Scribner, S., and Cole, M. (1978). Cognitive consequences of formal and informal education. Science, 182, 553-559.

Vygotsky, L.S. (1978). Mind in society. Cambridge, Mass: Harvard University Press.

The Effects of Orienting Strategies and Cognitive Processing Time on Factual and Inferential Learning

M.J. Hannafin, T.L. Phillips, L.P. Rieber, C. Garhart

The Pennsylvania State University

ABSTRACT

In this study, the effects of explicit versus general orienting strategies and varied amounts of processing time on the learning of facts and inferences were studied. A CAI lesson focusing on fictionalized science concepts was presented. Students were randomly assigned to either a cognitively or behaviorally based embedded orienting strategy group, receiving either 10 or 30 seconds of cognitive processing time to utilize the strategy provided. Results indicated that orienting strategy explicitness did not affect the learning of either facts or inferences differentially. A marginal effect was found for processing time, with students performing best with 30 seconds. The results suggest that differences in orienting strategies may not be as important as sufficient time for strategy utilization.

Extensive research has been done on the effects of orienting strategies on learning (Tennyson & Park, 1980; Mayer, 1979). An orienting strategy is a mediator through which new information is presented to the learner (Hannafin & Hughes, 1985). Some orienting strategies focus attention to specific factual information such as introductory behavioral objectives and specific pre-questions. Other strategies, such as advance organizers, tend to give more general guidance, facilitating the integration of new with existing information.

Learning outcomes may be effected differentially by the nature and position of various orienting strategies (Anderson & Biddle, 1975; Duchastel & Nungester, 1985). The effect of an orienting strategy is dependent partly on the specificity of the strategy. Behavioral strategies tend to focus attention toward specific information, increasing the learners abilities to process factual information but doing little to assist in the processing of inferential information (cf Mayer, 1984). Specific questions embedded throughout lessons have produced effective learning of intended concepts but limited incidential learning (Hannafin, 1984).

The use of cognitive strategies, however, may assist the learners in processing a more generalized version of instruction, thereby improving the learning of non-specific inferential information. Advance organizers, a type of cognitive strategy utilized to prepare the learner for new information, have been found to increase both comprehension and retention of information (Stone, 1983; Holzman, Allen, & Layne, 1982).

The amount of cognitive processing time provided during instruction also affects learning (Travers, 1982; Chu & Schram, 1975; Tennyson, Christensen, & Park, 1984). Cognitive processing time is the amount of time provided the learner to process and incorporate instruction into memory. Processing of informatin requires sufficient time to select, encode, and integrate (Tennyson, Christensen, & Park, 1984). When individuals are provided more information than able to process, they often become disorganized to the point of being unable to process information at all (Travers, 1982). Rest times between video segments, for example, may increase learning by providing information process time (Chu & Schramm, 1975). Unlimited processing time, as in the case of self-paced instruction, however, may be ineffective since learners may not attend to the information meaningfully for extended periods of time (Tennyson, 1984). Whereas providing limited processing time may inhibit learning, reductions of processing time may significantly debilitate learner performance (Travers, 1982; Belland, Taylor, Canelos, Dwyer, & Baker, in press).

Both orienting strategies and cognitive processing time affect learning. The purpose of this study was to examine the effects of orienting strategies and cognitive processing time on the learning of both factual and inferential information.

Methods

Subjects

The subjects were 40 college upper-division undergraduate and graduate students enrolled in computer literacy courses. Students participated during a regularly scheduled class session and were provided extra credit for participation.

Materials

The instructional treatments were modified versions of a CAI lesson which was developed for a previous research study (see Garhart & Hannafin, 1985). The lesson described the discovery of a fictitious element on a fictitious island, and the scientific, cultural and political ramifications of the discovery on a previously primitive society. The lesson was presented as a factual account of the discovery: students were not informed as to the fictitious nature of the lesson until the study was completed. This lesson was chosen for the following reasons: 1) the fictitious content was plausible; 2) the fictitious nature eliminated the effects of prior learning; and 3) the ease with which factual and inferential test items could be identified.

Three orienting strategies were developed: 1) embedded behavioral orienting strategies; 2) embedded cognitive orienting strategies; and 3) prompting to use individual learning strategies.

Behavioral orienting strategy. This strategy oriented the student with prompts which were specifically related to the factual content of the lesson segment which followed. The prompts provided the learner a specific orientation to the factual material subsequently assessed. The strategy consisted of a single computer frame, directing the student to attend to two specific informational items which included subsequent lesson text. One example of this strategy is: "In the next section be sure to learn this information: When the anthropologists began arriving on Jexium Island." These factual items consisted of names of people, places, specific events, and other details presented during the following lesson segment.

Cognitive orienting strategy. This strategy oriented the student to the instruction with a broader, more abstract method. These prompts were not tied to specific facts, but were designed to provide a broader contextual orientation to the content which followed. The strategy consisted of a computer frame directing the student to consider two general concepts which were to follow in the lesson text. An example of this strategy is: "In the next section, you will be presented information about: The importance of studying cultures."

Individual orienting strategy. This strategy provided no prompts to direct student attention to lesson information. Instead, the strategy simply advised the students to pay close attention to the information which followed. An example of this strategy is: "In the next section, try your best to learn the information." As in the prior two cases, three strategy frames were given before questioning.

All treatment strategies were embedded at identical lesson locations. Each strategy was presented in two processing versions: 10 seconds and 30 seconds. The strategy remained on the screen during this time and the computer ignored student input during the alloted processing time. The student was also given a general orientation as to the nature and organization of the lesson as well as directions for answering the questions on the computer.

The lesson consisted of four parts. Each part consisted of 10 to 14 text frames of easy to read, double-spaced paragraphs. Six graphic displays which supported the thematic content of the lesson were also included at various points, but were not related to specific criterion information. The orienting strategies were embedded at three evenly spaced intervals during each of the four parts.

Each part was followed by six questions: three factual type and three inferential. The 24 embedded questions also served as posttest items. Feedback was not provided after any of the questions during either the lesson or the posttest.

Each part of the lesson began with a banner page which was displayed for 3 seconds. The first orienting strategy frame was then displayed for either 10 or 30 seconds. This was followed by four to six frames of instruction. This sequence was repeated two additional times. The student was then presented a transition frame explaining that the lesson part had concluded and that six questions would follow. After answering these open-ended short answer type questions, the student was presented with the next banner frame for the next lesson part. This cycle was repeated for all four lesson parts.

At the conclusion of the lesson, the student was given a transition frame explaining the posttest. The posttest presented all 24 embedded questions, but in random order. The student was continually informed of the number of questions remaining in the posttest. Additional prompts were given one-third and two-thirds through the posttest. At the end of the posttest, the students were informed of the fictitious nature of the lesson and were directed to signal the proctor that the lesson was completed.

Dependent Measures

Embedded postadjunct questions. Each of the four parts of the lesson was followed by six postadjunct questions: three factual and three inferential. These were open ended, short answer questions. Factual questions measured recall of information presented during the lesson; the inference questions assessed the accuracy of conclusions based on lesson content. Reliability was .91 for the factual scale, and .81 for the inference scale. Validity was established through test item-lesson congruence and review by four evaluators.

Posttest. The 24 item posttest was a repetition of the four groups of six embedded questions presented in a random order.

Student response time. Time required by students to answer each of the two types of questions, adjunct and posttest, was also collected and collated by scale: factual and inferential. Response time was calculated by the computer and rounded to the nearest second.

Procedures

Students were randomly assigned to one of the six treatments upon arrival to class. All students were given an instruction sheet which was summarized briefly by the proctor. Each student was assigned a microcomputer terminal and given a computer diskette in accordance with their treatment group. Participants completed the lesson and posttest at their own rate, signaling the proctor when finished.

The study was conducted during three sessions spanning a four-day period. Students completed the study in times ranging from approximately 45 to 105 minutes. In order to avoid possible contaminatin between the sessions, students were briefed following their participation and urged not to discuss any portions of the lesson until completion of the study. Also, the students were randomly assigned to the treatments during each day of the study in order to randomize possible contamination effects over time.

Results and Discussion

Learning Effects

Marginally significant effects were obtained for cognitive processing time, $F(1,43)=2.92$, $p<.10$. Students provided 30 seconds to utilize the orienting strategy performed slightly better overall than those given 10 seconds. This effect was consistent across orienting strategy, although the magnitude of the effect was modest. Marginal differences were also found between en-route performance on embedded questions and the corresponding items on the posttest, $F(1,43)=2.62$, $p<.10$. The direction of this effect, however, was not anticipated. Student performance on posttest items was slightly better than on the embedded questions during instruction. Since the embedded questions did not include either feedback or remediation, this effect cannot be attributed to either correction or confirmation resulting from practice. Instead, the effect was likely attributable to the cueing function served by the question. En-route questions appear to cue students to presumably important lesson information. The inclusion of a question appears to direct students to retain the information contained in the question, while permitting them to ease cognitive overload by either forgetting or attending less to non-questioned information.

Although an orienting strategy main effect was not found, an orienting strategy-by-test scale interaction was detected, $F(2,43)=4.08$, $p<.05$. This interaction was characterized by better performance for the cognitive and behavioral strategies for factual versus inferential learning, while the individual strategy was most effective for inferential versus factual learning.

Response Time

The time required to respond declined significantly from the embedded to the posttest, $F(1,43)=96.65$, $p<.0001$. This may be due to the familiarity of students with the items contained in both embedded questions and posttest portions of the study. It is also possible that students were simply more confident of their responses during the posttest, and responded more rapidly.

Students also responded more rapidly to factual versus inferential questions, $F(1,43)=80.16$, $p<.0001$. The relative complexity of recall of inferential information could contribute to the observed differences. Vickers and Packers (1982) posited a cognitive complexity paradigm. Conclusions that involve greater evaluation of "below-surface" informatin integrated within existing cognitive networks require greater time to retrieve. This is likely to be the case for inferential tasks, where several pieces of learned information must be evaluated concurrently in order to form conclusions. Conversely, learning that is more explicitly defined, such as factual recall, requiring less time to retrieve (cf Vickers & Packers, 1982).

In addition, a test interval-by-test scale interaction was also significant, $F(1,43)=21.95$, $p<.0001$. This interactin was characterized by a regression toward the mean during the posttest. Inferential questions still required significantly more time to answer, but the differences in response time from embedded to posttest items was not uniform.

General Discussion

The results of this study suggest that providing more time for cognitive processing will generally improve learner performance. This appears to be true regardless of the type of orienting strategy present. Since only two processing times, 10 and 30 seconds, were studied here, it remains to be seen if this directional difference would continue beyond 30 seconds, and if so, how far. This is inconsistent with research by Belland, et al (in press), who suggested that reductions in allotted processing time tends to intensify effort and improve learning.

Another implication of this study pertains to the role of embedded questions as an organizational strategy. Embedded questions seem to provide an additional cue with which to organize and retrieve both factual and inferred lesson material. This contention is supported by the decrease in response time during the posttest. Since posttest questions were identical in form to the embedded questions, students were already familiar with the style and content of the posttest. It is also possible that students gained insight to questions answered incorrectly on the embedded questions as they continued through the lesson, thus correcting mistakes made during the lesson.

The increased time necessary to answer inferential questions is not surprising. Retrieval time for a question which require higher level cognitive tasks should generally be greater than for tasks which required only a lower level cognitive task, such as factual recall (Vickers & Packers, 1982).

The surprising result found in this study was the general lack of differential effect attributable to orienting strategies. It was hypothesized that students in the cognitive orienting strategy group would generally perform better than the other two groups, especially on inferential questioning. However, this was not found in this study. Carlson, Kincaid, Lance & Hodgson (1976) noted that students tend to revert to their own individual strategies regardless of how they are prompted during instruction. If this is the case, the orienting strategies would all assume the characteristic of a "use your own" strategy. This might account for why processing time resulted in more noticeable effects.

Several directions for further research are indicated. The effects of processing time needs further refinement in order to expand the contention that increases in processing time in general aids learning. Also, further research is necessary to determine if a ceiling level for processing time exists, and if so, the relationship between processing time limits and different cognitive tasks. Research is also needed to study whether or not students use imposed orienting strategies, or if they simply revert to individual strategies acquired over time. It would also be of interest to study developmental influences with young subjects, since they may not be as likely to have highly refined existing cognitive strategies to rely upon.

This study has raised several questions concerning how students learn from computer-based instruction. Based on this study, we can tentatively conclude that students require time to process instruction, and that the computer's potential for rapid access may need to be controlled to ensure adequate processing time. Further research should clarify the roles of orienting strategies and cognitive processing time in supporting learning.

REFERENCES

Anderson, R., & Biddle, B. (1975). On Asking people questions about what they are reading. In G. Bower (Ed.), Psychology of learning and motivation. New York: Academic Press, Vol. 9.

Belland, J., Taylor, W., Canelos, J. Dwyer, F., & Baker, P. (in press). Is the self-paced instructional program, via micro-computer based instruction, the most effective method of addressing individual learning differences? Educational Communication and Technology Journal.

Carlson, R. F., Kincaid, J. P., Lance, S., & Hodgson, T. (1976). Spontaneous use of mnemonics and grade point average. Journal of Psychology, 91, 117-122.

Chu, G. C. & Schramm, W. (1967). Learning from television: What the research says. Washington, DC: National Association of Educational Broadcasters.

Duchastel, P. C. & Nungester, R. J. (1985). Adjunct question effects with review. Contemporary Educational Psychology, 9, 97-103.

Garhart, C. & Hannafin, M. J. (1985). The accuracy of comprehension monitoring during computer-based instruction. Manuscript submitted for publication.

Hannafin, M. J. (1984). Guidelines for using locus of instructional control in the design of computer-assisted instruction. Journal of Instructional Development, 7(3), 6-9.

Hannafin, M. J. (1985). Empirical issues in the stury of interactive video. Educational Communication and Technology Journal.

Hannafin, M. J., & Hughes, C. W. (1985). Utilizing orienting strategies in the design of computer-based interactive video. Manuscript submitted for publication.

Holzman, T. G., Allen, S. A., & Layne, B. H. (1982). Advance organizers as aids for reading: Nature of the learner and the learning. Journal of Experimental Education, 50(2), 77-82.

Mayer, R. E. (1979). Can advance organizers influence meaningful learning? Review of Educational Research, 49, 371-383.

Mayer, R. E. (1984). Aids to text comprehension. Educational Psychology, 19(1), 30-42.

Stone, C. L. (1983). A meta-analysis of advance organizer studies. Journal of Experimental Education, 5(4), 194-199.

Tennyson, R. D., & Park, D. (1980). Teaching concepts: A review of instructional design research literature. Review of Educational Research, 50, 55-70.

Tennyson, R. D., Christensen, D. L., & Park, S. I. (1984). The Minnesota Adaptive Instructional System: An intelligent CBI system. Journal of Computer-Based Instruction, 11, 2-13.

Tennyson, R. D. (1984). Artificial intelligence methods in computer-based instructional design. Journal of Instructional Development, 7(3), 17-22.

Travers, R. M. (1982). Essentials of learning. New York: Macmillan Publishing Co., Inc.

Vickers, D. & Packers, J. (1982). Effects of attending set for speed or accuracy on response time, accuracy, and confidence in a uni-dimensional discrimination task. ACTA Psychologica, 50, 179-197.

THE IMPACT OF COMPUTER TECHNOLOGY ON PROGRAMMED INSTRUCTION MATERIALS IN SPEECH COMMUNICATION

Michael R. Hemphill
University of Arkansas at Little Rock

Christina C. Standerfer
University of Arkansas at Little Rock

ABSTRACT

A direct and identical comparison of public speaking programmed instruction and CAI materials found that students indicated more enjoyment and greater preference for CAI but took longer in completing the lesson. No differences were found between the two materials in regards to perceived validity, perceived difficulty, or learning.

RATIONALE

Effective human communication is recognized as the means by which we cope with an increasingly complex world. Colleges and universities have recognized the importance of effective communication skills by placing an increasing emphasis on its study. A recent survey of 552 member institutions of the Speech Communication Association found that over 40% of the schools described their basic speech course as focusing on a combination of public speaking, interpersonal communication, and small group communication -- topics and activities with both theoretical and practical application (Gibson, Gruner, Hanna, Smythe, & Hayes, 1980).

While such diversity broadens the students' exposure to the topic area, it is not without its disadvantages. Among the problems with the basic speech course reported by the schools in the survey were class size, inadequate time to cover course objectives, negative student attitudes toward the required course, lack of equipment and facilities, and a diversity of student interest. Any solution to these problems must preserve both theoretical and practical approaches, provide for the treatment of some topics outside of class (thus freeeing up more in-class time), while generating and/or preserving positive student attitudes toward the course. A common and often effective solution to these problems has been the use of a variety of self-instructional materials. One of these methods, programmed instruction (PI), allows the student to progress at his or her own pace and is typically completed outside of the classroom.

PI has been used frequently in the speech communication area (Amato, 1964; Boileau, 1983; Hanna & Gibson, 1983; Hubbard, 1983), but has not found favor with students and/or educators. Hanna and Gibson (1983) believe that by not incorporating computer technology at the outset, PI started out obsolete. The suggestion is that computer technology will somehow enhance the self-instructional experience beyond that provided by PI. In addition, Hanna and Gibson (1983) argue that PI, as a self-instructional strategy, is perceived by students as too simplistic and lacking validity as a means of delivering information.

Before organizational resources are devoted to the development of CAI materials for use in the speech communication classroom, some effort should be made to document its effectiveness. While previous research has shown that the use of CAI enhances the traditional instruction given in the classroom (Davis & Mount, 1984; Dossett & Hulvershorn, 1983; Hativa, 1984), no research to date has examined the impact of CAI in the speech communication field. Furthermore, while a previous study compared PI and CAI materials within a training context (Deignan & Duncan, 1978), no indication was given of how similar the designs of each set of materials were. No true comparison of the two types of delivery systems can be made unless the design of each is the same.

This paper is the initial study of a programmatic research effort designed to document those aspects of CAI design that are most effective in helping students communicate more effectively. As a starting point, this study compares traditional PI materials to a similarly designed CAI lesson in an attempt to assess the impact of the two delivery systems. More specifically, to what extent do CAI and PI materials differentially affect a student's: (1) motivation to learn, (2) perceptions of the lesson's validity, (3) perceptions of the lesson's difficulty, (4) knowledge of lesson information, and (5) time spent completing the lesson?

METHOD

88 undergraduates enrolled in an introductory speech course required of all students attending the university participated in this study. The students were required, as part of their coursework, to complete self-instructional study units on a number of different topics.

Two versions of a unit utilizing PI materials in public speaking (Markle & King, 1972) were prepared for this study. The PI version consisted of 63 fill-in-the-blank statements relating to speech preparation. The CAI version duplicated exactly the information contained in the PI version. Each fill-in-the-blank statement included in the CAI lesson was immediately followed by a duplicate screen which contained the correct answer. Thus, the PI version contained 63 statements, with correct answers identified in the margin, and the CAI version contained the exact same statements and answers on 126 screens.

Students were randomly assigned to receive either the PI version or the CAI version of the speech preparation materials. After completing the necessary requirements for the instructional unit, students completed a questionnaire containing items which measured the student's motivation, the perceived validity of the lesson materials, and the perceived difficulty of the lesson materials on a five-point scale.

In addition, students completed a seven question exam covering material in the lesson. Five items on the test were objective multiple-choice questions. The remaining items asked the student to list the major points of the lesson. Information retention was measured by percentage of correct responses to the multiple choice questions, and the number of correct items listed on the open-ended questions.

RESULTS

The research questions were examined by a simple comparison of the mean responses from the various dependent measures. An examination of the means revealed only three differences between the two groups. Students in the CAI group took significantly longer ($x=38.88$ minutes) than students in the PI group ($x=21.16$ minutes) to complete the lesson ($t=7.96$, $df=85$, $p < .001$). In addition, students in the CAI group indicated a more positive attitude toward the Study Center's use of CAI materials ($x=3.71$) and a greater preference for the use of CAI over in-class activities ($x=3.11$) than those students in the PI group ($x=3.02, 2.69$; $t=2.13, 1.90$ respectively, $p < .05$).

DISCUSSION

Few differences in the effects of similar CAI and PI speech communication materials were found in this study. Yet the results have very definite implications for those considering the use of computer technology in delivering instructional materials to students.

Students in the CAI group indicated more enjoyment associated with their lesson and a greater preference for CAI materials than did students in the PI group. Clark's (1983) meta-analysis suggested that positive attitudes toward CAI, such as those found in this study, may be attributed to the novelty of computers in the classroom. However, such positive attitudes may diminish over time as the student becomes accustomed to the technology (Cohen, 1977; Cohen, Ebling, & Kulik, 1981). Thus, educators should not be lulled by students initial positive attitudes into a false sense of CAI effectiveness.

The other major difference between the two modes of presentation in this study involved the amount of time required to complete the lesson. Previous research has suggested that one advantage of CAI is its efficient use of time (see, e.g., Deignan & Duncan, 1978). However, the results of this study suggest if a CAI lesson is designed exactly like a PI lesson, it will take the students longer to progress through the CAI lesson. The probable reason for this finding is the involvement which the computer forces upon the student. Unlike a textbook, the pages of which can be skimmed or skipped entirely, a CAI lesson requires a constant interaction with the student (even at the most superficial button-pushing level). The inability to move several screens ahead and the constant student activity with the lesson terminal means more time spent with the lesson.

This finding is especially interesting given the "less-is-more" philosophy of previous research examining the time issue. From a teacher's or trainer's point of view the less amount of time spent in a self-instructional setting, the more time available for other activities. It is obvious from this study that the mere presence of computer technology as a delivery system will not necessarily affect time in such an advantageous way.

The remaining comparisons of this study found no differences between PI and CAI in terms of perceived validity, perceived difficulty, and learning. These are important "non-finds" given the exact replication of the PI lesson on the computer. They suggest that few attitudes concerning learning and the outcomes of the process are strongly influenced by the mere presence of an advanced technology delivery system. Instead, it would appear that the design of the lesson is more important to attitudes and learning than the mode of lesson delivery. It may even be possible that a redesign of current PI materials may produce the same positive affects that are currently associated with CAI.

Developers and users of CAI have repeatedly emphasized the importance of design in creating effective lessons (Brownfield & Vik, 1983; Grillort & Govarts, 1984; Mahoney & Lyday, 1984). However, in the field of communication the plea has been to simply use computer technology to maximize the effectiveness of student-teacher interactions. But effectiveness apparently does not lie with the technology, it lies with the design of the lesson. This is what all educators, including those in the field of communication, must focus on if they are to develop truly effective self-instructional lessons.

This study represents an important first step in documenting the effects of using CAI in a discipline which currently relies on PI materials. The supposed disadvantages of PI will not be overcome by the mere substitution of a computer for a workbook. Rather, computer technology may offer the instructor a wider range of design strategies than PI ever could and therein lies the potential benefits.

REFERENCES

Amato, P. P. (1964). Programmed instruction: Its potential utility in speech. The Speech Teacher, 13, 21-24.

Boileau, D. M. (1983). Programmed instruction: A lull before the storm. Communication Education, 32, 137-143.

Brownfield, S. & Vik, G. (1983, December). Teaching basic skills with computer games. Training and Development Journal, pp. 53-56.

Clark, R. E. (1983). Reconsidering research on learning from media. Review of Education Research, 53, 445-459.

Cohen, J. (1977). Statistical power analysis for the behavioral sciences (Rev. ed.). New York: Academic Press.

Cohen, P., Ebling, B., & Kulik, J. (1981). A meta-analysis of outcome studies of visual based instruction. Educational Communication and Technology Journal, 29, 26-36.

Davis, B. L. & Mount, M. K. (1984). Effectiveness of performance appraisal training using computer-assisted instruction and behavior modeling. Personnel Psychology, 37, 439-452.

Deignan, G. M. & Duncan, R. E. (1978). CAI in three medical training courses: It was effective! Behavior Research Methods and Instrumentation, 2, 228-230.

Dossett, D. L. & Hulvershorn, P. (1983). Increasing technical training efficiency: Peer training via computer assisted instruction. Journal of Applied Psychology, 68, 552-558.

Gayeski, D. M. (1983, December). Making it all interactive. Training and Development Journal, pp. 40-42.

Gibson, J. W., Gruner, C. R., Hanna, M.S., Smythe, M. J. & Hayes, M. T. (1980). The basic course in speech at U. S. colleges and universities. Communication Education, 29, 1-9.

Govarts, K. C. & Gillot, T. A. (1984, November). Computer based industrial training: A primer. Training and Development Journal, pp. 28-31.

Hanna, M. S. & Gibson, J. W. (1983). Programmed instruction in communication. Computers and Education, 8, 293-304.

Hubbard, R. C. (1983). The use of programmed speech communication textbooks as a means of meeting the needs of heterogeneous students in large class sections. Communication Education, 32, 9-13.

Mahoney, F. X. & Lyday, N. L. (1984, July). Design is what counts in computer based training. Training and Development Journal, pp. 40-41.

Markle, M. K. & King, T. R. (1972). A program on speech preparation. Columbus: Charles Merrill Publishing Co.

THE COURSEWARE® INSTRUCTIONAL TOOLKIT:™
A Prototype Automated ISD System for Personal Computers

Greg Kearsley
Courseware® Inc.

Abstract

This presentation describes the results of an ongoing project to design and develop a prototype automated Instructional Systems Development (ISD) system called the Courseware® Instructional Toolkit . The purpose of the system is to explore potential productivity gains that might be achieved by an automated ISD system. A second goal of the project is to examine the extent to which ISD procedures can be automated and the research issues associated with automation. The current version of the system runs on an IBM PC/XT and is based upon commercially available applications software such as word processing, spreadsheets, database management, project scheduling and authoring programs.

Background

Instructional design and development is a highly labor-intensive process. Because of this, it is very expensive, prone to individual differences in skill levels, and not especially reliable. Despite the emergence and widespread use of Instructional Systems Development (ISD) methods and procedures, instructional design is still not a highly refined activity. There is a recognized need to improve the quality of ISD activities (e.g., Montague et al, 1983).

One obvious solution to the problem is to automate the instructional development process. Over the past ten years, a number of research projects have examined this possibility. Braby and Kincaid (1981) developed a computer editing system that helped authors create Navy technical manuals, and Kincaid et al. (1983) describe a system for editing test questions. O'Neal and O'Neal (1979) describe the Author Management System developed to automate project management for large-scale ISD projects. Brecke and Blaiwes (1981) described the CASDAT system which was intended to automate instructional analysis and design activities for technical training. Merrill and Wood (1984) describe their Lesson Design System implemented on an Apple II microcomputer.

A great deal of attention has been given to automating one particular aspect of instructional development, namely the creation of computer-based instruction programs. Authoring systems allow instructional programs to be developed without the need to know a programming language (Kearsley 1982; 1984). An obvious extension of authoring systems is to automate the analysis, design, and evaluation stages associated with the creation of courseware.

Indeed, if we look ahead a little, it seems fairly likely that almost all aspects of media production will be computer-based in one form or another. At the present time, inexpensive personal computers provide the capability to create, edit, and typeset text, or generate graphics, slides and overheads. More expensive computer systems allow the production of animated sequences. The inevitability of automated media production makes automation of the other aspects of instructional development even more attractive.

Benefits and Issues

The automation of ISD offers many

potential practical benefits including:

- Reduced time and costs to carry out ISD activities
- Increased standardization and consistency of ISD methods
- Improved quality of ISD results
- Better continuity and coordination in ISD projects

From a management perspective, a very important characteristic of an automated ISD system is that it allows much better control and tracking of an ISD project. To the extent that ISD activities are done on-line, it is possible for a manager to display the current status and results of any stage of a project. It also makes it easy to accumulate data on production time and costs to be used as a historical database for cost/time projections on future projects.

The automation of ISD also raises some interesting research issues. These include:

- What aspects of ISD can/cannot be automated?
- How will automation change the process and results of ISD?
- What are the relationships among different ISD activities?
- What effects will automated systems have on the skill levels of ISD personnel?

Probably the most significant theoretical implication of automated ISD is that it requires a very detailed understanding of instructional design and development. In order to write a program to carry out a procedure, it must be possible to state it explicitly. However, a lot of ISD methodology is currently based upon intuitive judgments and tacit knowledge. In order to automate such processes, intuition and tacit knowledge must be made explicit. Thus, one of the valuable outcomes of automating ISD is a more detailed understanding of creative steps in instructional development.

With all of these practical and theoretical considerations in mind, a research project was initiated by Courseware, Inc., to design and develop a prototype automated ISD system on an IBM PC/XT. This system was intended to build upon previous and concurrent automated ISD efforts at Courseware using various mainframe, minicomputer and word processing systems. The IBM PC was selected as the target machine because of its common availability and the large amount of software available.

Design Criteria

The Toolkit is intended to provide three major capabilities to the instructional designer:

- Database Management - The Toolkit should help a designer collect data for an ISD task, create a database from that data, manipulate the database in any way desired, and generate any reports desired.

- ISD Tutoring/Advice - The Toolkit should be able to provide help or tutorials to the designer on ISD methods or procedures. This is needed by both the novice and expert designer.

- Development Tools - The Toolkit should provide a range of software tools to assist in creation of graphics or multimedia materials, statistical data analysis, readability analysis, budgeting, etc.

Three major design criteria were followed:

- Open Architecture - The system was designed so that off-the-shelf application packages could be used. This avoids total reliance on custom-designed programs which require lengthy testing and debugging and provides a high degree of flexibility and extendibility.

- Interface That Accommodates Both Novice and Expert - The use of menus allows anyone to use the system. However, experts familiar with the particular application software involved can use the command language of the program instead of the Toolkit menus.

- Provision of Helps/Tutorials - Every function of the Toolkit should have both a help capability and a tutorial component. The helps provide advice on how to use a particular Toolkit procedure or program; tutorials explain an ISD method or activity.

The open architecture characteristic merits some additional discussion. By using off-the-shelf applications software, it is possible to develop and implement Toolkit applications very quickly. In addition, programming expertise is not required. Applications software can be written and modified by the users themselves (i.e., instructional designers). The full capabilities and flexibility of the applications software can be taken advantage of with minimum programming resources and development time.

However, a major limitation of relying primarily on off-the-shelf application software is that the ISD functions must be conceptualized in terms of the capabilities of the program. For example, if a database management or spreadsheet program is being used, the function must be created in terms of database or spreadsheet capabilities. In some cases, these built-in capabilities do not fit the problem domain very well and a custom-developed program is needed.

An additional consideration with the open architecture approach is that a considerable amount of time is required to evaluate and learn how to use applications programs. For a given application, time must be spent to determine what specific software program might be applicable. Even though the actual usage of a program may be straightforward, many hours of experience with the program may be needed in order to develop the application. Programming support may not be needed, but system analyst support will be required for this purpose.

Structure and Function of the System

The Toolkit prototype consisted of the three major components shown in Figure 1.

One of the most important components of the Toolkit is the system management software. This program manages and integrates all other programs in the Toolkit and acts as a pseudo-operating system. It captures and executes all keystrokes including function keys. It allows a uniform interface to be created for all programs. It provides data transfer from one program to another. It also provides windowing capabilities. The DESQ program from Quarterdeck Office Systems was selected as the system manager for the Toolkit prototype because of its early availability and reasonable costs. DESQ allows user defined menus to be created, features windowing and data transfer between windows, and can execute almost any DOS program. Other system managers for the IBM PC now available include Topview (IBM), Windows (Microsoft) and GEM (Digital Research).

Applications software includes any commercially available program for word processing, database management, spreadsheets, communications, etc. In the prototype, we used a variety of different programs including Lotus 1-2-3 (Lotus Development Corp), Wordstar (MicroPro), DBase II (Aston Tate), the Project Manager (Scitor), and others.

Authoring software could be any commercially available authoring language or system. This would include authoring languages such as PILOT or TenCore, and authoring systems such as PCIS (IBM/CSR), the Author (Raptor), SAM (Learn Com), Authority (ITS), etc. Graphics creation utilities, typesetting programs and printer utilities could also be included.

Custom designed software are programs that must be specially written in a general purpose language such as Basic, C, Pascal, Lisp, etc. These would be developed to accommodate a unique ISD project or requirement.

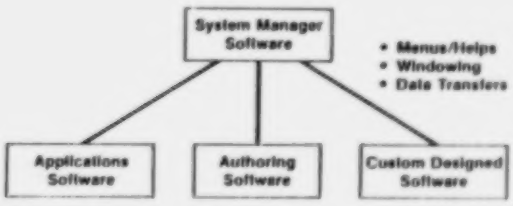


Figure 1. System Design

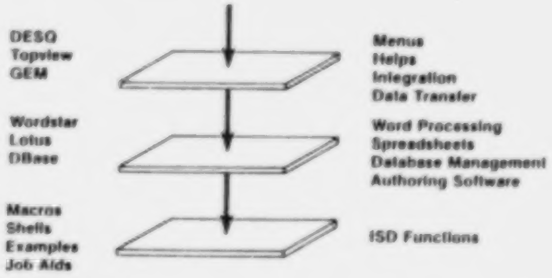


Figure 2. Functional Levels

Figure 2 summarizes the three functional levels of software that a user goes through when using the Toolkit. The system manager is largely transparent except when the user wants to manually transfer data from one window to another. The system manager presents menus (or helps) which list ISD functions, when the user selects an ISD function, the system manager calls the appropriate applications program, which in turn displays its own menus (or helps) local to that program. Users do not have to know how to use the applications program that a function is implemented in (although they can use the full capabilities if they want).

Toolkit Functions

Figure 3 depicts the specific ISD functions that were automated in the Toolkit.

An attempt was made to try and automate three or four major functions in each phase of ISD. This was done in order to provide enough scope in the prototype to investigate the interaction and integration of ISD functions across phases. The functions implemented also represent the most commonly performed ISD activities.

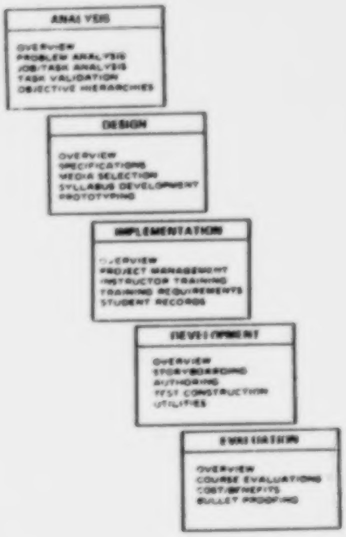

ANALYSIS
OVERVIEW
PROBLEM ANALYSIS
JOB/TASK ANALYSIS
TASK VALIDATION
OBJECTIVE HIERARCHIES
DESIGN
OVERVIEW
SPECIFICATIONS
MEDIA SELECTION
SYLLABUS DEVELOPMENT
PROTOTYPING
IMPLEMENTATION
OVERVIEW
PROJECT MANAGEMENT
INSTRUCTOR TRAINING
TRAINING REQUIREMENTS
STUDENT RECORDS
DEVELOPMENT
OVERVIEW
STORYBOARDING
AUTHORING
TEST CONSTRUCTION
UTILITIES
EVALUATION
OVERVIEW
COURSE EVALUATIONS
COST/BENEFITS
BULLET PROOFING

Figure 3. Courseware Instructional Toolkit

Each ISD function has a series of standard components: overviews, macros, shells, examples, job aids, and helps. The overview defines the nature and rationale of the ISD task and relates it to other tasks. Although the overviews in the prototype, were reference only, they could be interactive tutorials that actually teach the ISD task. Macros represent the actual automated capabilities implemented in the command language of the applications programs or the system manager. In the prototype they were always presented as a vertical or horizontal menu of keywords.

Shells represent an empty outline that can be completed (online or offline) to accomplish the particular ISD functions. For example, the shell for the lesson specification function is the outline of a lesson specification that must be filled in. Similarly the shell for storyboarding is a blank storyboard form. The shell for a cost analysis is an empty spreadsheet template.

Examples are shells which have been completed for specific instructional projects. They serve as models or illustrations of what the task looks like when completed. If desired, the designer could modify the information in an example to correspond to the current situation and hence, reuse the example.

Job aids are descriptions of the steps or procedures to be followed to complete an ISD function. In some cases the job is simply a shell with a listing of the questions to be asked to complete the shell. Once the questions have been answered, the function is complete.

The last functional component are helps. Helps explain how to use the current function. In some cases, this means explanations of what the macros do; in other cases, it means explanations of the ISD function itself.

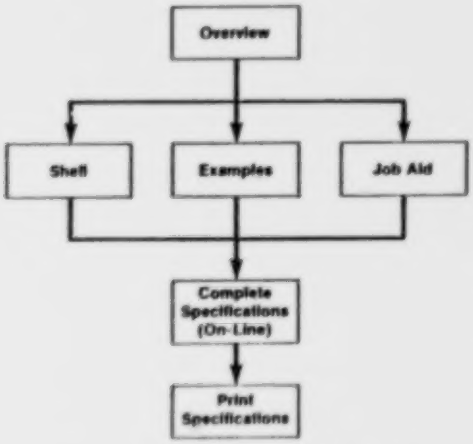

Overview
Shell
Examples
Job Aid
Complete Specifications (On-Line)
Print Specifications

Figure 4. Lesson Specification Example

BEST COPY AVAILABLE

A couple of examples should help clarify how a designer uses Toolkit. Figure 4 shows a schematic of how lesson specifications can be written.

The designer may or may not need an explanation of what lesson specifications are provided by the overview. The designer then has the option to select a shell and begin creating a specification, look at (or modify) some examples, or use a job aid. Using one of these three components, the designer would complete the lesson specifications online and them print them out for review. Of course, if the designer is working with project members who also have PCs, the completed specifications can be transmitted electronically via local or remote networks and do not need to be put in paper form.

Figure 5 shows another example for the syllabus development function. The figure depicts the major steps in creating a syllabus using the Toolkit.

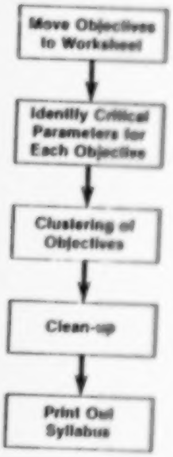

Move Objectives to Worksheet
Identify Critical Parameters for Each Objective
Clustering of Objectives
Clean-up
Print Out Syllabus

Figure 5. Syllabus Development Example

The instructional objectives are moved (automatically or manually) from the task listing functions onto a spreadsheet shell. This shell has column corresponding to the critical parameters that are used to arrange objectives into units and lessons (e.g., prerequisites, type of learning duration, etc.) Once these parameters have been identified for each objective, the program then clusters the objectives into units and lessons based upon rules about the parameters. The result is displayed to the designer who can then manually rearrange objectives or change parameters and rerun the program. Once the designer is happy with the syllabus it can be printed out (or transmitted) for review.

Conclusions

Because of time, budget and equipment limitations, it has not been possible yet to extensively field test the entire Toolkit in actual ISD projects. However, various Toolkit functions have been used individually in projects. The following conclusions are made on the basis of the initial testing of the system:

- Most ISD tasks consist of a relatively simple clerical component and a more complex judgmental component. The clerical component is easy to automate; the judgmental component is much more difficult and requires expert system type approaches.
- Automation could substantially speed up the ISD process. This increased productivity is primarily due to easy availability of examples, job aids, and templates which reduce the time to conduct a task. The computational and retrieval capability of a computer can also produce significant efficiency gains in some ISD activities.
- Transferring (or locating) information from one ISD task to another occupies a considerable amount of time. By automating the routine transfer paths (e.g., objectives to test construction; lesson specifications to storyboarding; evaluation data to lesson content, etc.), a significant amount of time can be saved.
- The easy availability of automated ISD functions encourages the use of those functions. ISD activities which are tedious or difficult are likely to be skipped. With the Toolkit, the probability is higher that all

tasks will be completed. Furthermore, if a selection of different approaches is provided for a particular task, the developer will think about which approach is best.

- The use of the Toolkit can increase standardization of ISD methodology. Because procedures are automated, they are carried out the same way each time they are used. This contrasts with manual procedures which are often changed each time they are used. Of course, a developer can always modify an automated procedure if necessary.
- The open architecture approach works and is much superior to a completely custom-programmed system because it is bug-free and can be modified by any user. Most ISD tasks can be handled by existing applications programs, but some tasks will require custom programming. However, these programs can be easily installed in the system to work in conjunction with off-the-shelf applications programs.

To summarize, the prototype version of the Toolkit has demonstrated that automating ISD on a personal computer is both feasible and highly desirable. Current efforts are focussed on implementing a fully operational version of the Toolkit based upon the lessons learned in the development and testing of the prototype.

Acknowledgement

The Instructional Toolkit Prototype was the joint effort of many Courseware, Inc. employees. In particular, Doreen Edwards, Nile Gardner, Diana Pickens, Susan Lindgren and Kathy Rose made significant contributions to its design and development.

References

Braby, R., & Kincaid, J.P., Computer aided authoring and editing. *Journal of Educational Technology Systems*, 1981-82 *10(2)*, 109-124.

Brecke, F., & Blaiwes, A., CASDAT: An innovative approach to more efficient ISD. *Journal of Educational Technology Systems*, 1981, *10(3)*, 271-283.

Kearsley, G., Authoring systems in computer-based education. *Communications of the ACM*., 1982, *25(7)*, 429-437.

Kearsley, G., Authoring tools: An overview. *Journal of Computer-Based Instruction*, 1984, *11(3)*, 67.

Kincaid, J.P., Braby, R.S., Wulfeck, W.H., Computer aids for editing test questions. *Educational Technology*, 1983, *23*, 29-33.

Merrill, M.D., & Wood, L.E., Computer guided instructional design. *Journal of Computer-Based Instruction*, 1984, *11(2)*, 60-63.

Montague, W.E., Wulfeck, W.H., & Ellis, J.A., Quality CBI depends on quality instructional design and quality implementation. *Journal of Computer-Based Instruction*, 1983, *10(384)*, 90-93.

O'Neal, A.F., & O'Neal, H.L., Author management systems. In H.F. O'Neil (Ed.), *Issues in Instructional Systems Design*. New York: Academic Press, 1979.

Cognitive Style and the Interpretation of Computer Graphics

John David Knechtel - Mohawk College

ABSTRACT

This study examines the relationship between field independence and performance in a computer maze, and concludes that field independence influences student's ability to decode computer graphics and to adjust to changes in orientation while in a computer maze. In this paper, educators are encouraged to examine the effect of level of field independence on student performance in computer-assisted instruction exercises.

This paper summarizes a study which investigated the performance of students in a computer maze as a function of cognitive style (Witkin & Goodenough, 1982). This study was undertaken to identify a specific situation for investigating the often made observation that students in a computer science class polarize into two groups: one group is enthusiastic about interacting with computers, another group is hesitant to interact with computers, often because they find the tasks required in programming and in operating computers difficult to understand. These two reasons reflect the interest of the author in a better understanding of the interaction between students and computers, and more generally, the interest of the author in the role of computers in education.

The "educational computing" literature reviewed for the study indicated that developments in computer technology in the decade preceding the study permitted the wide-spread use of computers in education, and that at the time of the study there were many unresolved issues associated with educational computing. It is the opinion of the author that implicit in these unresolved issues is the notion that research into instructional design and into the learning process has to proceed concurrently with the introduction of microcomputers in education. The articles of several other authors support this opinion (Braden, 1982; Montague et al., 1983). In the context of continuing the process of trying to understand the learning process and the role of computers in education, the author designed a study to confirm evidence in the literature that field dependent students should have more difficulty with computer tasks, if the computer tasks are similar to the perceptual tasks associated with the field independence construct. The implication of this finding is that if students are asked to complete educational assignments on a computer, and if these assignments require the student to interpret a computer graphics screen, then field dependent subjects should be at a disadvantage in these situations, not because they find the educational content difficult, but because they find the interpretation of a computer graphics screen difficult.

The population in this study was grade ten (10) boys from a high school in Ontario, Canada. The sample was obtained by first administering the Group Embedded Figures Test (Witkin et al., 1971) to the population. From the results of this test, two comparison groups were identified: one comparison group was the fifteen most highly field dependent boys (cut-off score of 17): the other comparison group was the fifteen most

Figure 1. Sample two dimensional overview from a computer maze.

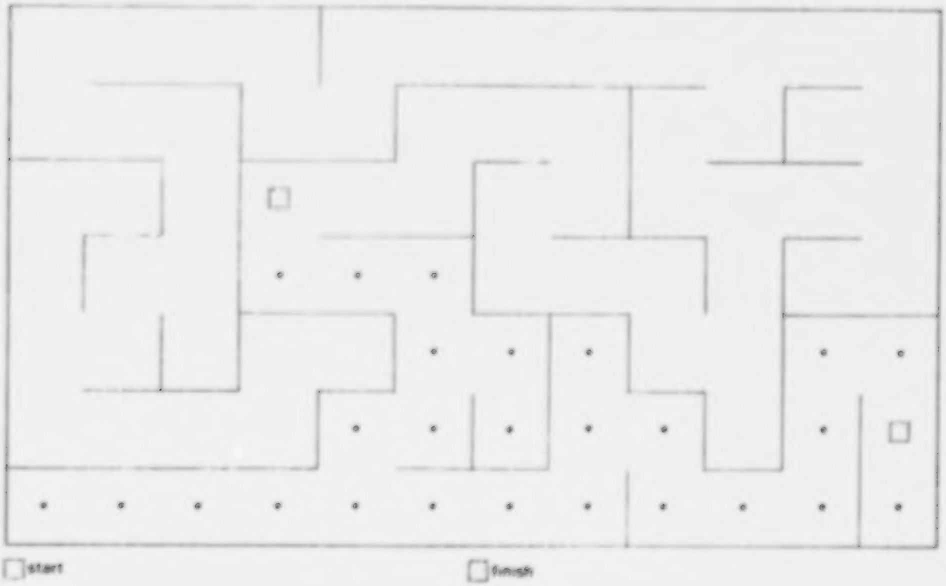


Figure 2. Sample three dimensional representations from a computer maze.

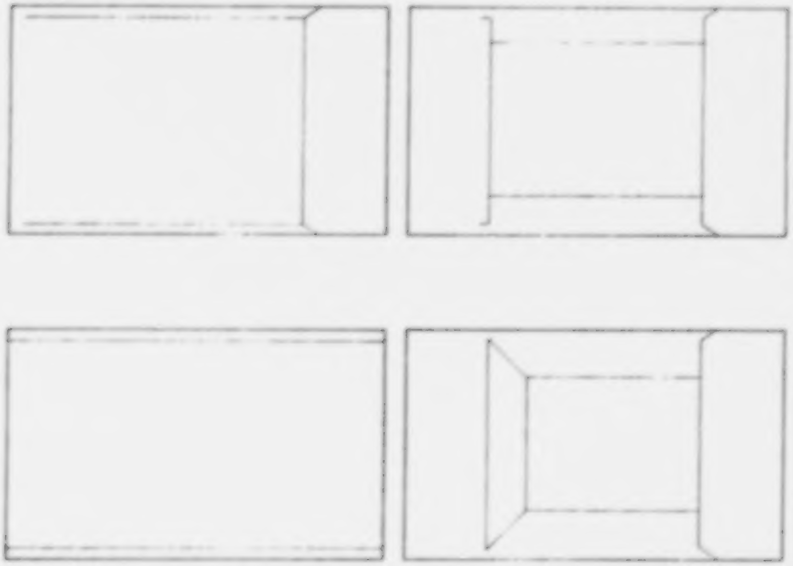


Table 1

List of Independent Variables and Dependent Variables

Independent Variables

Level of Field Dependence/Field Independence

Dependent Variables

Category	#	Dependent Variables
A: Decoding	1.	Trial Errors
	2.	Trial Response Accuracy
	3.	Trial Error Frequency
B: Fields of Reference	4.	Trial Orientation Errors
C: Points of Reference	5.	Trial Point of Reference Errors
D: Learning Style	6.	Trial Overviews
	7.	Average Overview Time
	8.	Trial Overview Frequency
	9.	Trial Assistance Level
E: Overall Performance	10.	Trial Time
	11.	Trial Moves
	12.	Trial Responses
	13.	Trial Economy of Moves
	14.	Trial Move Frequency

highly field independent boys (cut-off score of 5). It is important to note that the achievement levels of the two comparison groups at the time of the study were compared. Since there was no significant difference in the achievement levels of the two comparison groups at the time of the study, the author reasoned that any difference in ability to interpret the computer graphics was independent of academic achievement. The statistical evidence for this conclusion is reported in the original thesis document (Knechtel, 1984).

Each comparison group completed a training session and a microcomputer exercise. In the training session the nature of the task that the students completed on a microcomputer was explained. In the microcomputer exercise, the students were required to complete two computer mazes. To complete the mazes, subjects were required to interpret two and three dimensional diagrams on a computer screen. The two dimensional diagrams in the computer mazes (Figure 1), showed the halls of a computer maze from the perspective of a person standing over a maze and looking down at it. The three dimensional diagrams in the computer mazes (Figure 2) represented the halls which were visible to a subject while simulating a walk through an imaginary maze.

While the students completed each of the computer mazes, the responses of the students were captured by the microcomputer. After the completion of the two mazes, the sequence of responses by each student was reconstructed, and the time required by each subject to complete each response was determined. By this method, the data was collected, and the values for each dependent variable (Table 1) were calculated .

From the data collected from the first computer maze, three of the hypothesized differences between the two comparison groups were confirmed (Table 2). First, field dependent subjects had statistically more difficulty interpreting three dimensional diagrams from the computer maze. Second, field dependent subjects had statistically more difficulty with adjusting to changes in orientation in the computer maze. And third, the overall performance of field independent subjects on two variables was greater than the overall performance of the field dependent subjects. The complete statistical analysis for this study is recorded in the original thesis document (Knechtel, 1984). It is important to note that the results on the second computer maze completed by the two comparison groups were not significantly different. The implication of this finding is also reported in the original thesis document.

Table 2

Summary of Wilcoxon Rank Sum Tests - Dependent Variables - Maze #1

Dep. Var. No.	N		Mean of the Ranks	Sum of the Ranks		Critical Value
	Field Indep. Subjects	Field Dep. Subjects		Field Indep. Subjects	Field Dep. Subjects	
Decoding						
1.	15	14	210	178.5	163.5	171 *
2.	15	14	210	166.0	269.0	171 *
3.	15	14	210	160.5	262.5	171 *
Fields of Reference						
4.	15	14	210	162.5	184.0	171 *
Points of Reference						
5.	15	14	210	199.5	235.0	171
Learning Styles						
6.	15	14	210	183.5	198.5	171
7.	15	14	210	180.0	255.0	171
8.	15	14	210	201.5	216.5	171
9.	15	14	210	194.0	209.0	171
Overall Performance						
10.	15	14	210	164.0	179.0	171 *
11.	15	14	210	181.5	196.5	171
12.	15	14	210	166.0	181.0	171 *
13.	15	14	210	200.5	234.5	171
14.	15	14	210	206.0	221.0	171

* $p< .05$

Three points summarize the importance of this study. First, the literature reviewed by the author provided strong evidence that the role of computers in education is not well understood. Second, the author recognized from this study that examining of comparison groups selected on the basis of psychological tests as the groups interact with computers under controlled conditions does increase one's understanding of the interaction between students and computers. And third, the author has concluded from this study that ability to decode the three dimensional representations in a computer maze and ability to adjust to changes in field of reference in a computer maze is related to level of field independence as measured by the Group Embedded Figures Test, and related to ability to complete embedded figures tasks as measured by the Group Embedded Figures Test. It should be noted, however, that the author was unable to ascertain that the overall performance of field dependent subjects in this study was lower that the overall performance of the field independent subjects. The rationale for this last statement is reported in the original thesis document (Knechtel, 1984).

The conclusions from this study should be interpreted with caution for two reasons. First, the conclusions are based on non-parametric statistics. Non parametric statistics were necessary because of the nature of the data. However, non-parametric statistics hold less power than their parametric equivalents. And second, the conclusions should be generalized to other populations with caution since the sample was drawn from one school. Repetition of this study with a broader population, and with a variety of computer graphic situations would be useful since these studies would ascertain if the conclusions of this study apply to a broader population, and to a variety of computer graphics situations.

It should be remembered, however, that the greatest significance of the study is its implication: if students are asked to complete educational assignments on a computer, and if these assignments require the student to interpret a computer graphics screen, then field dependent subjects may be at a disadvantage in these situations, not because they find the educational content difficult, but because they find the interpretation of a computer graphics screen difficult. And since the difference in achievement scores for the two comparison groups at the time of the study was not significant, the reported differences between the two comparison groups appears to be independent of achievement of the subjects.

References

Braden, R. (1982). Instructional Design: the Microcomputer Connection. Educational Technology, 11, 30-31.

Knechtel, J.D. (1984). Performance in a Computer Maze as a Function of Cognitive Style. Unpublished Master's Thesis, University of British Columbia, Vancouver, B.C. Canada.

Montague, W. et al. (1983). Quality CBI depends on quality instructional design & quality implementation. Journal of Computer-based Instruction, 10(3) & 10(4), 90-93.

Witkin, H.A., & Goodenough, D.R. (1982). Cognitive Styles: Essence And Origins. New York: International Universities Press, Inc.

Witkin, H.A., Oltman, P.K., Raskin, E., & Karp, S.A. (1971). Manual for the Embedded Figures Test, Children's Embedded Figures Test, Group Embedded Figures Test, Consulting Psychologists Press, Inc., Palo Alto, California.

AN EMPIRICAL COMPARISON OF MENU CHOICE MECHANISMS FOR EDUCATIONAL COMPUTING SYSTEMS

Elaine Lewis, Ph.D.

Laboratory for Instructional Technology at Boston University
Boston MA
02215

ABSTRACT

This paper reports a study on the relative effectiveness of textual versus iconic menu formats during concept selection while using an educational computer system. Menu format has been correlated with user speed, selection accuracy, and user satisfaction. Seventy-two subjects were individually evaluated. Findings indicate that subjects made menu selections more quickly using the text version with no significant difference in accuracy. Some difference in satisfaction was due to menu format; participants were more likely to respond that the graphics enhanced the program in the icon version. When findings were considered according to native language, selection accuracy was significantly greater for non-native speakers of English while using the iconic menu.

THEORY AND RATIONALE

Developers of interactive systems have come to realize that greater knowledge of user psychology is crucial (Miller and Thomas, 1977). In order to make intelligent decisions about screen design and interactive techniques, software designers must know how the medium will affect the user. They need empirically based standards of quality.

Although there is a well-established literature on the most effective use of more traditional media for educational purposes (Salomon, 1979), relatively little is known about standards for educational computing systems. This research offers empirical evidence about effective menu choice mechanisms for educational software.

Menu Variables

Textual menus have been in common use for decades. Consequently, there is a large body of research which addresses the design of word-based menus. Studies have considered organization, item order, target size and positioning in terms of how they affect users' speed and accuracy (Snowberry, 1983; Card, 1982; Merrill and Bunderson, 1979). There is little research that compares textual menus with more visual approaches, although common knowledge suggests that an iconic approach has some benefits for beginning users.

Often graphic symbols, or icons, can represent meanings more immediately than the equivalent text string, making iconic menus more compact than text menus for some applications. Icons are especially good for representing objects or simple actions (i.e. scissors for cutting, garbage can to delete). Several popular computer systems, such as the Apple Lisa and Macintosh, and the Xerox Star currently use iconic menus for objects and attributes. Bewley and associates at Xerox have performed timed recognition and naming tests for alternative icon designs used by Star (Bewley, *et al*., 1983). How an iconic format compares directly to a textual format has yet to be determined.

This project directly compares the relative effectiveness of textual versus iconic menu formats for educational software. Menu format is considered a categorical variable with these two options, it is correlated with user speed, selection accuracy, and user satisfaction.

User Variables

Human factors literature suggests three primary criteria for the quality of an interactive information system: 1. the time it takes a user to accomplish a particular task; 2. the accuracy with which the user can accomplish the task; and 3. the comfort of the user while doing the task (Landauer, et al., 1982). These criteria are the bases for the user variables: speed, accuracy, and satisfaction.

These variables have been operationalized as follows. Selection SPEED is measured as the amount of time in seconds necessary for the user to complete the entire program. ACCURACY is measured in terms of number of points achieved for correct selections minus penalties for incorrect responses and information clues. SATISFACTION has been evaluated through psychographic measures in the form of semantic differential attitude scales on an exit questionnaire. Satisfaction scales include "easy to use" versus "hard to use" and other bipolar adjectives and descriptive phrases. Speed and accuracy are considered ratio level variables. Satisfaction uses an interval scale to interpret the semantic differentials.

A number of user characteristics were also measured through the exit questionnaire. These variables included standard demographics such as age, educational level, and occupation, along with computer-related training and experience. Other factors which could affect verbal versus pictorial information processing such as native language and handedness were also considered.

METHODOLOGY

Research Design

This study employed an experimental method under controlled laboratory conditions. The experiment has a between group design comparing the categories of menu presentation format. Thirty-six subjects were assigned to each of the experimental cells, for a total of 72 subjects. The user variables speed, accuracy and satisfaction were measured for all research participants. Variable levels were compared between the conditions and tested for significance.

Research participants were volunteers drawn from the Boston University community. University faculty, staff, and students are considered suitable subjects because they represent a typical population likely to use educational computing systems. Non-native speakers of English were actively recruited in order to assess any differences due to language factors. Since all subjects were either University staff (primarily faculty and clerical) or currently enrolled in courses, it is assumed that all subjects have a reading comprehension level in English adequate to understand instructions and skill definitions.

Stimuli and Procedure

Stimuli were two interactive instructional modules which addressed the experimental conditions. Content was consistent for each; menu format varied in terms of using textual or iconic presentation mechanisms.

Both modules began with a brief introductory segment which set the scenario. Subjects were asked to play the role of producer for an interactive videodisc program. Then subjects were required to learn the icon or keyword for six production crew members along with a description of each member's responsibilities and talents. Finally, subjects were confronted with six dilemmas, each could be solved through one of the crew members. The identification task involved choosing the appropriate crew member by touching the keyword or icon which represented that person. Subjects received one hundred points for each correct identification, lost thirty points for each incorrect response, and could request a clue by paying twenty points. The entire procedure took about ten minutes.

During the experiment, individual subjects were asked to view stimuli using one of the interactive workstations with touch sensitive screens available in Boston University's Laboratory for Instructional Technology. In each case a research assistant was present to ensure consistency in experimental procedure.

After a brief introduction to the system which included an explanation of the experimental procedure, each subject was asked to view one of the modules and follow all instructions. Elapsed time (includes subject decision-making time only, does not include computer plotting time) and score were coded by the system. After finishing the module, subjects were asked to complete the satisfaction variable questionnaire.

RESEARCH FINDINGS AND INTERPRETATION

Subject Characteristics

Seventy-two subjects completed the test, thirty-six used the icon version and thirty-six the text version. Subjects had an average age of 27 years; fifty percent were college graduates and nearly thirty percent hold a graduate degree. Sixty-six percent were full or part-time students; twenty-six percent worked full time. Sixty percent were native speakers of English (n=44) and forty percent have native languages other than English (n=28).

Subjects have had more experience with computers than the general population. When asked how often they use a computer, twenty-six percent responded "never", thirty-six percent indicated "under ten hours per week", eighteen percent "ten to nineteen hours" and nearly twenty percent over twenty hours per week. Twenty-five percent responded that they had a computer at home, and over seventy percent have had at least one computer related course.

These characteristics imply that findings can be generalized most safely for adult learners who have had some experience with computers.

Differences Due to Stimulus Type

The three dependent variables speed, accuracy, and user satisfaction were operationalized through number of seconds to complete the program, number of points achieved, and five differential scales for satisfaction. The scales used to describe the program in general were interesting/uninteresting, easy to do/difficult to do, and fun to do/boring to do. Two scales referred to the design and graphics used in the program: poor design/good design and graphics enhanced program/detracted from the program.

T-tests were used to assess differences for all variables according to the two experimental conditions. Although subjects scored higher while using the icon version and completed the test more quickly with the text version, average score and average speed according to group were not statistically significant. Results for score and speed are shown by Table 1.

		mean	std dev	T	prob
SCORE	ICONS	485.556	81.537	0.50	0.61
	TEXT	474.167	108.269		
SPEED	ICONS	269.333	90.701	1.81	0.07
	TEXT	232.861	79.602		

degrees of freedom = 70

Table 1: T-tests for SCORE and SPEED

Examination of T-tests which compare user satisfaction for the two menu formats revealed no statistically significant differences for four of the semantic differential scales. One scale, "Did the computer graphics and design enhance or detract from the program?" (1=enhance, 7=detract) showed a difference significant at the 0.01 level. Subjects who interacted with the icon version of the program were more likely to say that the graphics and design enhanced the program. Full results are shown in Table 2.

		mean	std dev	T	prob
INTRST/UNINTRSTING	ICONS	1.78	0.96	-0.23	0.82
	TEXT	1.83	0.92		
EASY/DIFFICULT	ICONS	2.26	1.31	-0.17	0.87
	TEXT	2.31	1.49		
FUN TO DO/BORING	ICONS	2.03	1.23	-0.63	0.53
	TEXT	2.20	1.07		
POOR DSN/GOOD DSN	ICONS	6.19	1.14	1.26	0.21
	TEXT	5.83	1.29		
GRPH ENHANCE/DTRCT	ICONS	1.69	0.78	-3.22	0.002
	TEXT	2.71	1.71		

degrees of freedom = 69

Table 2: T-Tests - Satisfaction Variables

Differences Due to Native Language

When data were analyzed according to subjects' native language more significant findings emerge. When considering the entire sample, for both stimuli according to native language, non-native English speakers did nearly as well in terms of score, but had significantly slower completion times. These results are shown in Table 3.

When the language groups are considered separately according to stimulus type, a strong relationship is shown. Native English speakers score somewhat higher with the text version, while non-native speakers score

		mean	std dev	T	prob
SCORE	ENGLISH	491.136	77.736	1.16	0.25
	NON-ENG	462.142	117.232		
SPEED	ENGLISH	212.818	49.336	-4.90	0.00
	NON-ENG	311.250	98.733		

degrees of freedom = 42
(separate variance estimate)

Table 3: SCORE - SPEED by Native Language

significantly higher with the icon version. Both groups completed the text version more quickly, although this trend is not significant for the non-native group. These findings are shown by Tables 4 and 5.

		mean	std dev	T	prob
SCORE	ICONS	473.479	84.885	-1.62	0.11
	TEXT	510.476	65.687		
SPEED	ICONS	241.391	43.263	5.08	0.00
	TEXT	181.523	34.744		

degrees of freedom = 42

Table 4: T-Tests for SCORE and SPEED for Native English Speakers

		mean	std dev	T	prob
SCORE	ICONS	506.923	73.528	2.06	0.05
	TEXT	423.333	135.629		
SPEED	ICONS	318.769	128.136	0.35	0.727
	TEXT	304.733	68.036		

degrees of freedom = 26

Table 5: T-Tests for SCORE and SPEED for Non-Native English Speakers

CONCLUSIONS AND FUTURE RESEARCH

While it is dangerous to speculate based on the results of a single study, this experiment suggests several areas fertile for future research. It lays the foundation for research questions like: "How does the representational quality of an icon affect performance?" and "Do people less fluent in their native English language prefer iconic menus?"

Nevertheless, this study has yielded evidence for the common-sense notion that iconic menus may be superior when your audience contains non-native speakers of English. Future studies may provide more refined guidelines for designers of computer-based educational systems.

BIBLIOGRAPHY

BEWLEY, W., T. Roberts, D. Schroit, and W. Verplank, "Human Factors Testing in the Design of Xerox's 'Star' Office Workstation", Proceedings of CHI83 Human Factors in Computing Systems, ACM, 1983, pp. 72-77.

CARD, S.K., "User Perceptual Mechanisms in the Search of Computer Command Menus", paper presented to Conference on Human Factors in Computing Systems, Gathersburg, MD, March 1982.

FOLEY, James D., Victor L. Wallace and Peggy Chan, "The Human Factors oF Computer Graphics Interaction Techniques", IEEE Computer Graphics and Applications, November 1984, pp. 13-43.

LANDAUER, T.K., S.T. Dumais, L.M. Gomez, and G.W. Furnas, "Human Factors in Data Access", The Bell System Technical Journal, vol. 61, no. 9, November 1982, pp. 2487-2509.

MERRILL, Paul F. and C. Victor Bunderson, "Guidelines for Employing Graphics in a Videodisc Training Delivery System", First Annual Report of ISD for Videodisc Training Systems, Volume III, Army Research Institute for the Behavioral and Social Sciences, Alexandria, Virginia, 1979.

MILLER, L.A. and J.C. Thomas, "Behavioral Issues in the Use of Interactive Systems", International Journal of Man-Machine Studies, vol. 9, 1977, pp. 509-536.

SALOMON, Gavriel, Interaction of Media, Cognition, and Learning, SanFrancisco, Jossey-Bass, 1979.

SNOWBERRY, Kathleen, Stanley R. Parkinson, and Norwood Sisson, "Computer Display Menus", Ergonomics, vol. 26, no. 7, 1983, pp. 699-712.

SMark: AN AUTOMATED SYSTEM FOR GRADING SHORT ANSWERS IN NATURAL LANGUAGE

Martin Janta-Polczynski, Gerard K. Rambally, Arjinder S. Uppal

Department of Computer Science, University of Regina
Regina, Saskatchewan Canada. S4S 0A2

ABSTRACT

A computer grading system for evaluating short-answers written in a Natural Language is discussed. To this point, the program has graded answers written in English and French. The evaluation process involves two stages: First, the key-concepts in correct answers from experts are identified. The student's answer is then processed to determine which of these concepts are present. Secondly, a mark is assigned, depending on which combination of key-concepts occur in the student's answer. The theoretical basis for the model used to assign a mark, on the basis of identifying the key-concepts is inspired by the theory of **rough sets**.

1. INTRODUCTION

In previous approaches to natural language processing, much care has been taken to provide a precise syntactic (Sager, 1981) or semantic (Charniak, 1985; Simmons, 1984) representation. In this research project, on the other hand, we want a simplified, trans-semantic representation, that ignores linguistic and representational complexity in order to get to the "bare bones" of a short answer, and still be able to classify it as adequate or not.

This paper describes a computer program for automatically grading short answers (e.g. the definition of a technical term) written in a natural language. To this point, the program has graded answers written in English and French. The automated grading program, SMark (Smart Marker), which is implemented on a VAX 11/750 is written in ICON (Griswold, 1983), and involves two main stages in the evaluation of a student answer. The first stage (discussed in section 2), involves identifying the key-concepts in correct answers from experts. The student answer is then processed to determine which of these concepts are present. In the second stage (discussed in section 3), a mark is assigned, depending on which combination of key-concepts occur in the student answer.

2. PROCESSING NATURAL LANGUAGE TEXT

The objective of this processing stage is to identify and isolate the key concepts contained in the expert and student answers. Consider File 1 consisting of expert definitions of the term **multiprogramming**:

The file undergoes the following steps:
(i) Preprocessing: In this step all apostrophes are eliminated. E.g. **can't** becomes **not can**.

(ii) Hyphenation: Word phrases representing one concept are automatically hyphenated. E.g. **at the same time** becomes **at-the-same-time**.

(iii) Lexical analysis: In this step words are stripped of standard endings to get root words. E.G. **execution, executing** and **executes** all become **execute**.

(iv) Replace synonyms: The file is then updated from three synonym files (Logical, General and Specific [to the given definition]). E.g. **run** would be recognized as a synonym of **execute**, in the context of this specific definition.

(v) User-defined synonyms: The corpus of words of the expert file with their respective frequencies is output (grammatical words are ignored). The user may provide any synonyms or implied concepts.

(vi) Post-hyphenation: Once again automatic hyphenation is done on the file. In addition the user has the option of hyphenating any word phrases referring to one concept. If a new hyphenation is done by the user, these new hyphenated words are subsequently added to the file "Hyphenated.words".

(vii) Remove grammaticals: Finally all grammatical words (propositions, pronouns, articles, common adverbs, etc.) are removed from the definitions file.

The processed file is illustrated in File 2.

(viii) Frequency table: Finally a corpus from File 2 with its frequency is produced (see table 1). From this table the key-concepts are selected for grading purposes.
Note: the file of expert definitions has to go through the above process only once (at set-up time).

Similarly, the student-answers file also goes through the above process. Consider the following student answer for the term **multiprogramming:**

-> Original answer: **is when more than one program is running on a computer system.**

-> Hyphenated answer: **is when more-than-one program is running on a computer-system.**

-> Key-concepts after processing: **many program execute computer.**

Now, the student answer file is read to be graded and hence is passed on to part 2 of the program (see section 3).

3. MARKING PROCEDURE

Assume that students are asked a short-answer question (e.g. give an explanation or define a technical term, etc.) in a specific domain, and this answer is defined in terms of several key-concepts, where the student knowledge about the question is characterized in terms of the combination of key-concepts present in his/her answer. Further assume that some combinations of these concepts will constitute one or more correct answers; other combinations will yield partial answer. The objective is to categorize the student answers as being either:

-> Correct : in which case full marks are given,

-> Incorrect : in which case no marks are given,

-> Roughly correct : in which case partial marks are given.

The theoretical basis for the model used to grade natural-language short answers, on the basis of identifying the key-concepts is inspired by the theory of "rough" sets introduced by Pawlak (1984).

The basic component of the grading system is a set of complete and partial answers. Knowledge about these answers is expressed through the use of key-concepts which characterize the answers. Values of the concepts (0 or 1) represent whether the concept is present or not. Future, more sophisticated versions of SMark will include more values such as concept modifiers (**many, not, like, every**, etc.). Moreover, we assume that a function f assigning concept values to answers can be computed.

Formally, a Grading System S is a quadruple S=(A,C,V,f) where:

A - is a finite set of answers

C - is a finite set of key-concepts, obtained by processing a number of correct answers

V - is a set of values for the concepts (in the present case 0 or 1)

f - is a grading function and

f : A x C -> V such that f(a,c) V can be computed for all a A and c C

This Grading System may be represented as a table in which columns are concepts, and rows are answers and an entry in the a-th row and c-th column has the value f(a,c). In other words, each row in the table represents a complete answer (e.g. a_1 and a_2 in table 2) or a partial answer (e.g. a_3, a_4 and a_5 in table 2).

Recall that we want to classify the student answer as **correct, incorrect,** or **roughly correct.** If the student answer contains all the concepts in any one of the complete definitions, then the student answer is **correct.** E.g. **Multiprogramming is two or more programs running on the same computer simultaneously.**

Otherwise, the best we can hope for is to obtain an approximation of the expert answer. That is, the student only knows some concepts (or none!) contained in the answer. To deal with this case, we apply the notion of approximation of sets; specifically, the operations of **lower approximation** and **upper approximation** of an answer. These will determine the limits of tolerance for approximating one of the answers of the system by the student answer.

Let X be the key-concepts in a student answer. The **upper approximation** of X in S (denoted **SX**) is the smallest union of key-concepts in answers in S, containing the set X. A **lower approximation** of X in S (denoted $\underline{S}X$) is the largest union of key-concepts in answers in S contained in the set X.

Recall that if the student answer matches one of the complete answers then the student answer is **correct**. Otherwise we calculate the **lower** and **upper** approximations. If the **lower approximation** is empty (i.e. $\underline{S}X=\emptyset$) then the student answer is **incorrect** i.e. the student is not able to give even a partially correct answer. If the **lower approximation** is not empty (i.e. $\underline{S}X\neq\emptyset$), then the student answer is **roughly correct** because the student answer has some key-concepts which partially answer the question.

In order to assign a mark to a **roughly correct** answer the following procedure is followed. First, each of the key-concepts is assigned a normalized weight (see table 2) according to two criteria, namely: the frequency with which experts use the concepts and whether a concept is absolutely necessary for defining the term. Secondly, the **lower** and **upper approximations** of the student answer are determined. Finally a mark is assigned according to the following formula:

4. RESULTS and DISCUSSION

In a final examination for the introductory computer science class at the University of Regina, students were asked to define the following terms: **multiprogramming, computer network, register** and **system flowchart**. The answers were graded by a human grader and by the computer grading system, SMark. The results are shown in Table 3.

As is evident from Table 3, every question on every student paper was not assigned the same mark by the marker and SMark. Consider the question on **multiprogramming**: Thirty-three students answered this question and 21 of them were assigned an identical mark by the marker and SMark. The authors thoroughly examined the 12 mismatches (i.e. the answers which were assigned a different mark by the marker and by SMark) and found that in 11 of these cases the computer grader, SMark, was correct. The marker was either inconsistent or inaccurate in his grading. In the twelfth case, both SMark and the marker were wrong. The marker assigned a mark of 2, while Smark assigned a mark of 0, however the authors agreed on a mark of 1. Thirty-five students answered the question on **computer network**. Twenty-six of these answers were assigned an exact mark by the marker and SMark. On examination of the 9 mismatches, it was found that the computer grader was correct in all cases. There were only 4 mismatches of the 29 students who answered the question on **register**. An examination of the student answers in these 4 cases revealed that the computer grader was correct in 3 cases. In the fourth case, the marker assigned a grade of 2, while SMark assigned a grade of 1. The authors agreed with the marker.

$$\text{Mark} = \frac{\text{normalized weights of concepts in } \textbf{lower approximation}}{\text{normalized weights of concepts in } \textbf{upper approximation}} * \text{total marks for question}$$

E.g. Student answer: **Multiprogramming is programming more than one program at same time on a computer.**

Key-concepts in student answer:	**many**	**program**	**same-time**	**computer**
Weights of concepts :	0.203	0.203	0.203	0.185

Lower approximation : 1 1 0 1 0 (i.e. a_3 from Table 2)

Upper approximation : 1 1 1 1 1 (i.e. a_1 from Table 2)

$$\textbf{mark} = \frac{0.203 + 0.203 + 0.203}{1.0} * 2.0 = 1.222 \text{ rounded to } 1$$

Finally, on the question regarding **system flowchart**, there were 19 mismatches. The mark assigned by SMark was correct in 14 of these cases; the marker was correct in 3 cases, and both assigned an incorrect mark in 2 cases.

5. FUTURE DEVELOPMENTS

In rough-set theory, we try to make set-membership of an object correspond to specific combinations of attribute-values assigned to that object in an information-system. In our scheme, we decided the most likely attributes would be some kind of concepts. These key-concepts (as we called them) are derived a posteriori from the corpus of all correct answers by selecting those concepts with a high enough frequency (this threshold frequency has to be determined experimentally). As for the concepts themselves, we started with three different notions:

(1) **Concept-level$_0$**: Class of equivalence of synonymous terms
e.g. Many = {**more-than-one, two-or-more, several, multiple, set-of,** ...}

This notion has been successfully implemented. But in very rare cases, two definitions can have indistinguishable representations: We picked as our example **multiprogramming** which in terms of **Concept-level$_0$** is nearly indistinguishable from **multiprocessing**, but which was marked successfully by SMark nonetheless.

(2) **Concept-level$_1$**: It would be a **Concept-level$_0$** with a possible modifier, such as: **many, not, like, some,** etc.

(3) **Concept-level$_2$**: this would be a **Concept-level$_0$** expressing a relationship plus a **Concept-level$_0$** expression an entity. E.g. the definition of **flowchart** could contain a **Concept-level$_2$**: **describe + logic.**

To implement (2) and (3) would require goal-directed search, and would considerably complexify the system, which up to now ignores syntax entirely and is fed its semantics and expertise for the given definition by an expert-user. However once the system has processed a sufficient number of short-answers and corresponding marks, it could handle an unlimited number of other answers, provided the occasional new word can be fitted by the human expert in the already defined categories.

6. CONCLUSION

At the University of Regina, as is the case at most North American universities, enrollment in introductory computer-science classes has escalated to an overwhelming level. This, compounded with the shortage of qualified faculty, results in very large introductory classes. One implication of this is that the grading of mid-term and final examinations becomes a long and tedious process. It is the authors' expectation that SMark will alleviate some of this tedium by grading the questions on examinations which involve short natural-language answers with a high density of expected concepts. From the results to date, the evidence indicates that SMark will also perform this grading task with greater efficiency, accuracy and consistency than human graders.

Together with this pragmatic value, this research project has demonstrated a rather significant cognitive result. That is, in evaluating natural-language text, it suffices to identify the key-concepts used and not be too concerned with their order, their syntactic category, the grammatical and "empty" words connecting them, or any additional information adduced. In most cases it is reasonable and empirically validated (as shown in our test runs) to assume that if a student uses the correct concepts, he/she will use them with the proper logical organization (relation-words with correct arguments, acceptable ordering and appropriate connecting words). Any additional information (concepts), whether correct or not, is ignored and will not influence the grading.

Mandell (1983) : is a technique whereby several programs are placed in primary storage at the same time.
Long (1984) : is pertaining to the concurrent execution of two or more programs by a single computer.
Gore & Stubbe (1984) : is concurrent execution of segments of two or more programs by a computer.
Bitter (1984) : simultaneously running two or more programs on a computer system.
Parker (1984) : the simultaneous execution of two more program sequences by multiple computers oprating under common control.
Sanders (1983) : the simultaneous handling of multiple independent programs by interleaving or overlapping their execution.
Sumner (1985) : a system that has the ability to handle many programs at one time.
Spencer (1982) : running two or more programs at the same time in the same computer.
Harold (1984) : a process whereby several programs are placed in primary storage at the same time giving illusion that they are being executed simultaneously.
Stern & Stern (1982) : ability of a computer system to execute two or more programs simultaneously.
Shelly & Cashman (1980) : the concurrent execution of two or more computer programs on one computer system.
Capron & Williams (1982) : is the process of loading two or more programs into main storage and executing them concurrently.

File 1. Expert Definitions of the Term, **Multiprogramming**

Mandell : technique many program execute computer same-time
Long : same-time execute many program one computer
Gore & Stubbe : same-time execute sequence many program computer
Bitter : same-time execute many program computer
Parker : same-time execute many program sequence many computer execute common control
Sanders : same-time execute many independent program interleave overlap execute
Sumner : computer able execute many program same-time
Spencer : execute many program same-time one computer
Harold : execute many program execute computer same-time give illusion execute same-time
Stern & Stern : able computer execute many program same-time
Shelly & Cashman : same-time execute many program one computer
Capron & Williams : execute execute many program computer execute same-time

File 2. Processed File of Expert Definitions.

EXPERTS -->	1	2	3	4	5	6	7	8	9	10	11	12	
common	.	.	.	.	X	.	.	.	.	.	.	.	8 %
technique	X	.	.	.	.	.	.	.	.	.	.	.	8 %
control	.	.	.	.	X	.	.	.	.	.	.	.	8 %
independent	.	.	.	.	.	X	.	.	.	.	.	.	8 %
overlap	.	.	.	.	.	X	.	.	.	.	.	.	8 %
give	.	.	.	.	.	.	.	.	X	.	.	.	8 %
illusion	.	.	.	.	.	.	.	.	X	.	.	.	8 %
interleave	.	.	.	.	.	X	.	.	.	.	.	.	8 %
sequence	.	.	X	.	X	.	.	.	.	.	.	.	16 %
able	.	.	.	.	.	.	X	.	.	X	.	.	16 %
one	.	X	.	.	.	.	.	X	.	.	X	.	25 %
computer	X	X	X	X	X	.	X	X	X	X	X	X	91 %
program	X	X	X	X	X	X	X	X	X	X	X	X	100%
many	X	X	X	X	X	X	X	X	X	X	X	X	100%
same-time	X	X	X	X	X	X	X	X	X	X	X	X	100%
execute	X	X	X	X	X	X	X	X	X	X	X	X	100%

Table 1: Frequency listing for 'multiprogramming'

	many	program	execute	same-time	computer
a_1	1	1	1	1	1
a_2	1	1	1	1	0
a_3	1	1	0	1	0
a_4	1	1	1	0	0
a_5	1	1	1	0	1
weights(a_1)	0.203	0.203	0.203	0.203	0.185
weights(a_2)	0.250	0.250	0.250	0.250	0.000

Table 2: Grading System for the term **multiprogramming**

	Number of students who answered the question	Human Grader nbr. of students getting a mark of			Computer Grader nbr. of students getting a mark of			Number of exact matches between both types of graders			Total nbr. of mis-matches
		0	1	2	0	1	2	0	1	2	
Multiprogramming	33	13	2	18	11	9	13	9	0	12	12
Computer network	35	2	4	29	9	3	23	2	2	22	9
Register	29	7	4	18	10	3	16	7	2	16	4
System Flowchart	43	8	7	28	10	16	17	3	5	16	19

Table 3: Comparison of Human Grader and SMark

REFERENCES

Bitter, Gary G. (1984) **Computers in Today's World**, New York, John Wiley & Sons.

Capron, H.L., Williams, Brian K. (1982) **Computers and Data Processing**, Menlo Park, California, Benjamin/Cummings Publishing Co.

Charniak, Eugene and McDermott, Drew (1985), **Introduction to Artificial Intelligence**, Reading, Massachusetts, Addison-Wesley Publishing Co.

Gore, Marvin R., Stubbe, John W. (1984) **Computers and Information Systems**, New York, McGraw-Hill.

Griswold, Ralph E., Griswold, Madge T. (1983) **The ICON Programming Language**, New Jersey, Prentice-Hall.

Harold, Fred G. (1984) **Introduction to Computers with BASIC**, St. Paul, Minnesota, West Publishing Co.

Long, Larry (1984) **Introduction to Computers and Information Processing**, Englewood Cliffs, New Jersey, Prentice-Hall.

Mandell, Steven L. (1983) **Computers and Data Processing Today**, St. Paul, Minnesota, West Publishing Co.

Parker, Charles S. (1984) **Understanding Computers and Data Processing: Today and Tomorrow with BASIC** New York, Holt, Rinehart and Winston.

Pawlak, Z. (1984) **Rough Classification**, International Journal of Man-Machine Studies, 20, 469-483.

Sager, Naomi. (1981) **Natural Language Information Processing**, Reading, Massachusetts, Addison-Wesley Publishing Co.

Sanders, Donald H. (1983) **Computers Today**, New York, McGraw-Hill.

Shelly, Gary B., Cashman, Thomas J. (1980) **Introduction to Computers and Data Processing**, Brea, California, Anaheim Publishing Co.

Simmons, Robert F. (1984) **Computations from the English**, Englewood Cliffs, New Jersey, Prentice-Hall.

Spencer, Donald D. (1982) **Data Processing, An Introduction with BASIC**, Columbus, Ohio, Charles E. Merrill Publishing Co.

Stern, Robert A., Stern, Nancy (1982) **An Introduction and Information Processing**, New York, John Wiley & Sons.

Sumner, Mary (1985) **Computers Concepts and Uses**, Englewood Cliffs, New Jersey, Prentice-Hall.

BETTER CHEMISTRY THROUGH ICAI--GETTING STARTED

Lynn H. Smith
Roland Garton

Office of Computer-Based Instruction
University of Delaware

ABSTRACT

After one year's work, the authors have developed an early version of an artificially intelligent lesson which drills and remediates chemistry students on Lewis structures. This paper describes how the lesson will operate when complete, what the early versions will accomplish, what steps were taken, and what decisions were made to get the project to its current stage.

INTRODUCTION

Recently the University of Delaware decided to expand its Computer-Aided Instruction (CAI) efforts into the realm of Intelligent Computer-Aided Instruction (ICAI). Several projects were funded for the first time in 1984. One of those projects was to create a lesson that would allow chemistry students to practice drawing Lewis structures. Those of us assigned to the Lewis structure project searched the literature and found much of value. Nowhere, however, did we find a chronicle of the difficulties of starting out in ICAI. This paper is offered in the hope that our first year's experience with ICAI will be helpful to others just beginning to delve into this vast and fascinating field.

THE NATURE OF THE DOMAIN

The ability to predict how chemicals will react is a critical one for the chemist, and drawing Lewis structures is the first step in making the prediction. Simply put, a Lewis structure is a representation of a group of atoms, such as a molecule, showing the bonds between atoms and the electrons around the atoms. For example, a molecule of water, H_2O, is represented as shown in Figure 1. The lines between the H and O atoms represent single bonds. The dots around the O represent lone pairs of electrons about oxygen.

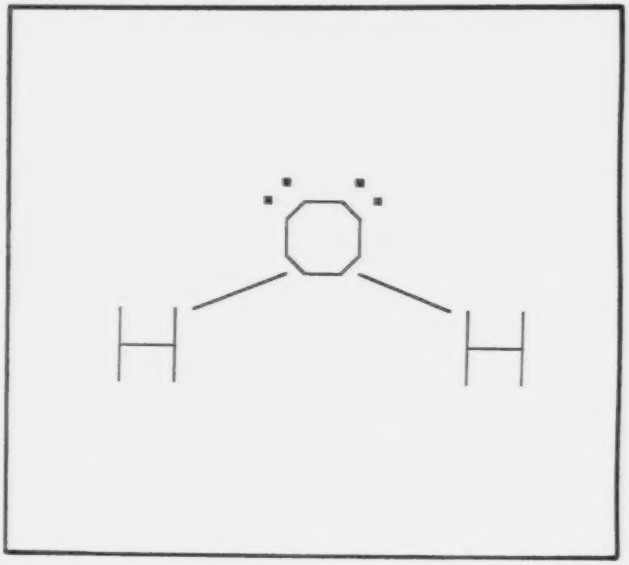


Figure 1
A Lewis structure for H_2O.

Once a Lewis structure is produced, the three-dimensional shape of a molecule can be determined. When the shape of a molecule is known, the strength and orientation of its overall charge and its polarity can be calculated. From these, the chemist can accurately predict how one compound may react with others and what products will result from the combining of given chemicals. With an incorrect Lewis structure at the beginning, the rest of the process is faulty, which could produce catastrophic results if the wrong chemicals were combined.

Solving Lewis structures is a rule-driven procedure. Like many expert domains, some of the rules permit exceptions. Developing expertise in solving these problems involves knowing which rules to apply and which exceptions to permit. The problem of teaching Lewis structures calls for an approach that makes an attempt to figure out where in the problem solving process the student erred, and then provides the kind of guidance appropriate for an individual student's errors.

WHY ICAI

The process of drawing Lewis structures lends itself well to ICAI in that it has a fairly limited and well-defined set of rules. Another advantage is that very little cultural or "external" knowledge is required to solve Lewis problems, as there would be in solving, say, word problems in mathematics. With a limited knowledge domain, Lewis structures offer a comfortably-sized experimental field.

One of the main advantages that ICAI brings to the lesson is in its ability to solve problems both using correct expert rules and student mal-rules. This is accomplished by separating the list of correct and incorrect rules from the routines that process the rules to arrive at the answer. Thus, rules can easily be added or deleted, just as a student does during the learning process. In this way, the computer builds a model of the student. Using this model the lesson can provide the most appropriate guidance, be it instruction about the rules, intense practice on specific problem types, or other remediation.

A GENERAL MODEL OF AN ICAI SYSTEM

In making a survey of ICAI literature, we developed a general model of an ICAI system, seen in Figure 2. This general model consists of a system of five interconnected modules, a student interface, an expert model, a teaching model, a student model, and a database.

The student interface is the part of the lesson that students see. In some cases, this is an existing lesson. For example, Brown and Burton developed an artificially intelligent coach for the game How The West Was Won (Sleeman and Brown, 1982, pp. 79-98; Barr and Feigenbaum, 1982, pp. 254-260), and I. P. Goldstein developed an AI coach for the game WUMPUS (Sleeman and Brown, 1982, pp. 51-78; Barr and Feigenbaum, 1982, pp. 261-268). In other cases, such as ours, the interface must be developed as part of the project.

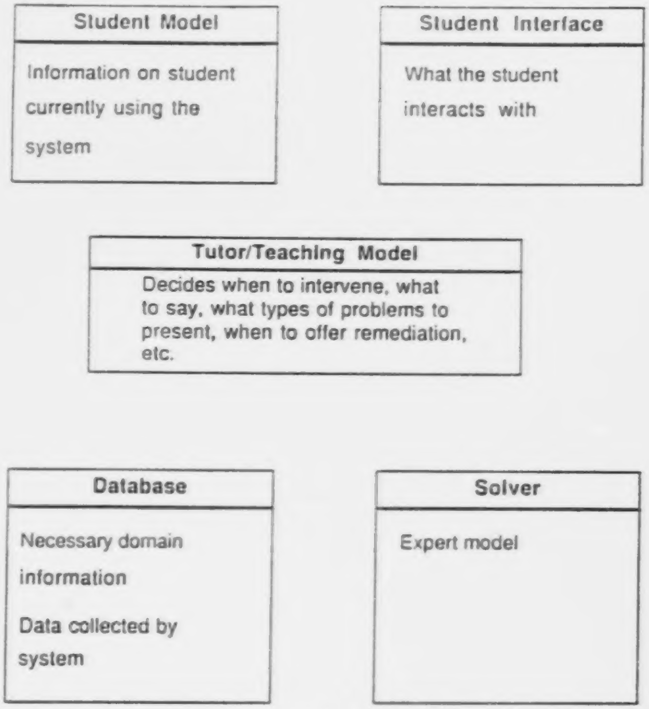

Student Model
Information on student currently using the system
Student Interface
What the student interacts with
Tutor/Teaching Model
Decides when to intervene, what to say, what types of problems to present, when to offer remediation, etc.
Database
Necessary domain information
Data collected by system
Solver
Expert model

Figure 2. General model of an ICAI system.

The expert model is the part of the ICAI system that can solve the problem posed to the student. An expert model may be the core of an ICAI system--the first part to be developed. For example, MYCIN (Sleeman and Brown, 1982, p. 202; Barr and Feigenbaum, 1982, pp. 192-198) is an expert system developed to aid doctors in the diagnosis and treatment of infectious diseases. Since this expert system proved difficult to use, the GUIDON (Sleeman and Brown, 1982, pp. 201-226; Barr and Feigenbaum, 1982, pp. 267-278) ICAI system was added to teach the use of the MYCIN expert.

More recently, the teaching model has been separated from the expert model, partially as a result of experiences like the MYCIN system, where the expert in the subject matter is not necessarily an expert in teaching. The WEST coach is another example where the rules for helping the student (teaching) are separate from the expert rules on playing the game.

Most ICAI systems contain a student model--a record of what difficulties the student is having. Burton's BUGGY (Sleeman and Brown, 1982, pp. 157-184; Barr and Feigenbaum, 1982, pp. 279-282) and IDEBUGGY (Sleeman and Brown, 1982, p. 173) systems contain a list of about 110 known bugs that students could possibly exhibit. The student model is built by identifying which bug or bugs each student has.

In our model of a general ICAI system we have placed the database as a separate module which can be accessed as needed by other modules in the system. The design of the database should be considered separately, rather than letting it grow as a haphazard collection. Since more than one of the other modules may access this data, it is important that the structure of the data be consistent and easily accessible.

PROPOSED LEWIS DOT LESSON

Figure 3 shows an idealized Lewis structure lesson. It contains all five modules described as part of an ICAI system. In this idealized Lewis structure lesson, the student interface will present the problem and allow input either by keyboard or by mouse. It will display the appropriate feedback or remediation, and it will contain a tutorial on how to use the system.

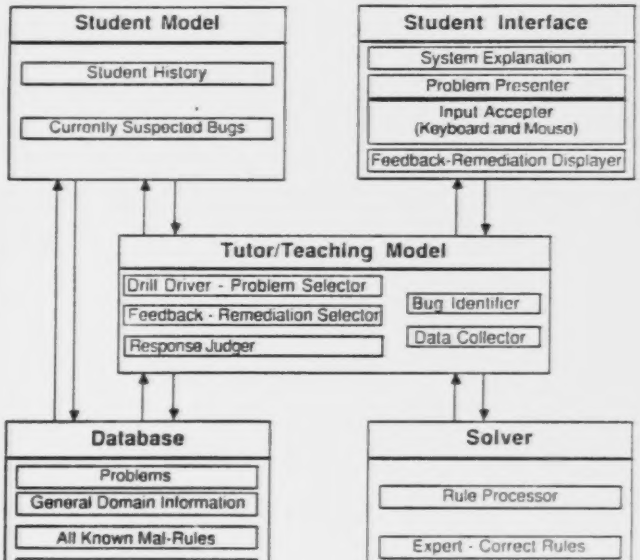

Student Model
Student History
Currently Suspected Bugs
Student Interface
System Explanation
Problem Presenter
Input Accepter (Keyboard and Mouse)
Feedback-Remediation Displayer
Tutor/Teaching Model
Drill Driver - Problem Selector
Feedback - Remediation Selector
Response Judger
Bug Identifier
Data Collector
Database
Problems
General Domain Information
All Known Mal-Rules
All Known Process Errors
Student Records
Research (Cumulative) Data
Solver
Rule Processor
Expert - Correct Rules

Figure 3
Idealized Lewis structure lesson.

The student model will consist of a student history, such as which problems the student has completed, whether each was correct on the first try, if not, what type of error was made, and how many times that problem was judged wrong. It will also contain a list of bugs that the student has.

The solver will contain a rule processor and a list of the correct (expert) rules, and it will keep a record of the process by which it obtains a solution. The database will contain a list of problems, in the form of formulas and a list of the general domain information, such as symbols and atomic numbers. Student records and cumulative research data will also be collected in this database. Finally, the database will contain a list of all known process errors and all known mal-rules (incorrect rules).

The heart of the idealized Lewis structure lesson is the tutor/teaching model. It functions as the control module of the lesson. As a tutor, it deals with how and when to present feedback and/or remediation. To do this, it must contain a response judger to compare the student's answer with the answer from the solver. A good tutor is also able to identify the bugs that the student exhibits, so a bug identifier is a component of the tutor. Once the bug is identified, appropriate problems must be presented to the student. If no single bug is identifiable, diagnostic problems must be selected. Finally, some component must control which data gets sent to the database.

As Figure 3 shows, most of the communication within the system goes through the control module, the tutor. The only exception to this is that the student model must be able to directly access the lists of known process errors and known mal-rules to build the list of suspected bugs.

PROJECT STRUCTURE

Creating an intelligent Lewis structure lesson requires expertise in several areas: chemistry, chemistry instruction, cognitive science, artificial intelligence systems, programming, and lesson design for CAI. We have gathered bits and pieces of expertise known by different people. The staff currently consists of a half-time professional programmer aided by two part-time student programmers, a lesson design expert working about four hours per week, two ICAI expert advisors and a chemistry professor serving as consultants, and a manager. As with any team effort, ideas and viewpoints are manifold, and molding those into a unified direction demands considerable time in coordination alone.

The need for close coordination is greatest when laying the initial groundwork. In fact, where a CAI lesson may take weeks in mapping strategy and outlining the content and approach, the Lewis structure lesson has taken months, and the time spent in developing and understanding the underlying theories still continues to be significant. One of the greatest differences we have discovered so far between CAI and ICAI development is the extent to which the knowledge domain must be known. To develop an expert system or to model a student making errors requires a great deal of time in mapping the real problem-solving process. For these and other reasons, the complexity of an ICAI lesson translates into a lengthy learning curve, especially in a project where ICAI is a new area for many of the personnel. In the nine months of project operation, the team has produced a pilot version of the lesson. Getting to the pilot stage, however, involved extensive discussion to arrive at basic decisons.

DEVELOPING THE LESSON

The first two decisions we had to make in this project were the choices of hardware and language. For language, some dialect of LISP seemed to be the obvious choice. For hardware, we had three possibilities for which LISP is available--a VAX 11/780, a Xerox Dandelion, or an IBM PC.

One thing we had decided to do was to use the computer to collect data for the project. This made our choice simple, since the only one of the above machines that would allow us to deliver the lesson to significant numbers of students was the IBM PC.

Since Lewis structures are drawn, we knew we would need graphics in the lesson. We found an implementation of LISP that had graphics capabilities available for the IBM PC, but the package required a significant portion of memory. We could not predict how much more space the LISP program itself would take. Furthermore, reports from other users indicated that the IBM PC implementations of LISP ran very slowly, so we considered other alternatives. Both Pascal and C are flexible languages, and much less memory-intensive than LISP in their IBM implementations. Our programmers were more familiar with C than with Pascal, and felt confident of being able to write LISP-like functions in C. These factors, programmer familiarity with C and the unknown eventual size and speed of LISP, led us to choose C as the delivery language.

One thing which facilitated this decision was the fact that we still had access to the Dandelion to experiment with approaches in LISP. Once a piece of LISP code worked the way we liked, it could then be translated to C. So far, the utilization of this has been minimal but we anticipate greater need for it as the project increases in complexity.

Once the hardware and language decisions were made, we had to tackle decisions about the lesson. With the resources and personnel we had, we could not write all of the modules at once.

We elected to develop the student interface first for several reasons. First, we were committed to helping students with Lewis structures as soon as possible. By developing a drill-and-practice type lesson, we could give them some help more quickly. Second, since only one member of the team other than the subject matter expert had experience with solving Lewis structures, developing the interface first allowed other team members to learn some of the subject matter. Third, having the interface would facilitate the data collection needed to develop both the student model and the expert model. Fourth, in developing a limited judger for the interface, we would begin work both on the data structures necessary for the whole system and on the rules for the expert module.

Figure 4 shows only the components of the ideal Lewis structure lesson that existed in the pilot lesson.

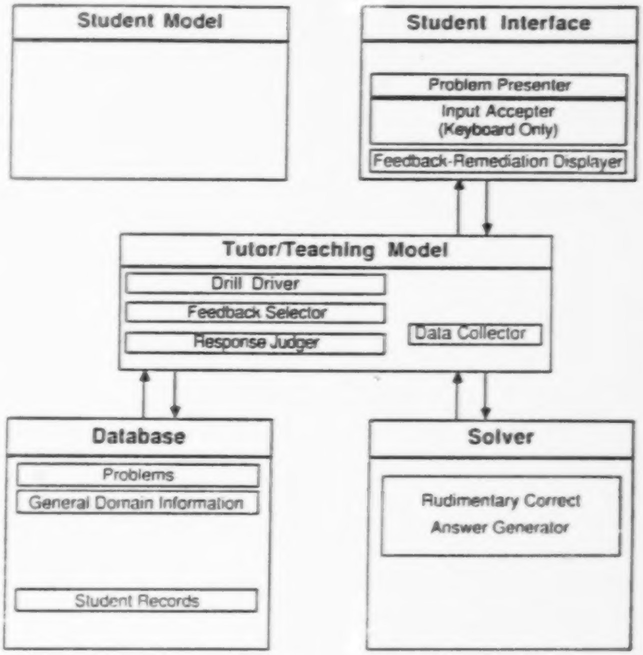


Figure 4
Lewis lesson at first pilot.

The drill driver in the pilot lesson is primitive, presenting problems randomly from three difficulty levels that were assigned by the chemistry professor. The data collector in the pilot lesson simply collects every student keypress into a file in the database, creating a "tape" of the student session. The database also contains a list of problems with the associated information that the judger needs, plus a list of general domain information.

The most important data structure in our system is the Lewis dot structure itself. The student will create a picture on the screen. The expert will have a formula and general domain information available. Both of these must be translated into the same data structure for comparison by the judger. The data structure must be set up so that different orientations or screen locations used by the student result in the same data structure. It must also be capable of being translated into a picture on the screen to show the student.

The response judger functions by making two passes. In the first pass it builds a data structure from the student's input while checking for obvious errors in the process (such as the wrong number of electrons). If no errors are detected in the first pass, then the data structure of the student's answer is compared to a data structure built from the problem data in the database.

The student model does not exist in the pilot lesson. The student interface has keyboard input (but not mouse input). The solver is not tied into the system at all in the pilot lesson. It consists of a preliminary list of rules which reside in the response judger.

FUTURE DECISIONS

The pilot lesson was tested with about twenty students in late June of 1985. The results of that test will aid in choosing which parts of the lesson will be worked on for the first revision.

Tentative plans have been made to test a revised lesson with about 150 students in the fall of 1985. Obviously, it is impractical to collect every keypress of 150 users. One improvement in the revised lesson must be the selection of appropriate data to collect. Since the response judger builds a record of the student's response as it moves through the judging process, we anticipate being able to save these records at preselected points. Evaluation of the data "tapes" from the pilot test will help in choosing the best points. For each problem, the pilot lesson also records the number of errors, classified into expected types. While this is valuable data, even more valuable will be the collection of unanticipated errors. This type of data will play a large role in creating the expert, the list of mal-rules, and the feedback and remediation. Once we have a more complete set of errors, cumulative data records will be helpful in improving the problem selector. Eventually, the problems will not be grouped by "difficulty level," but will have information on the types of errors that are most frequently associated with them. Using this information, the tutor can select problems for diagnosis, explanation, or remediation. This data will also be useful for research purposes.

Another improvement to the pilot lesson will be the addition of the ability to use a mouse for the input. The lead programmer is anxious to tackle the solver module and the manager wants to see some progress on the student model. Figure 5 shows one possible structure for the first revision lesson.

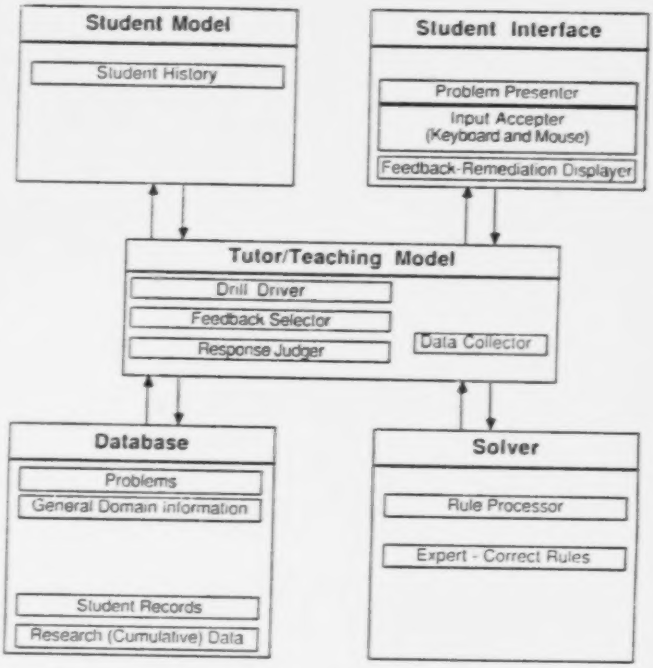


Figure 5. Possible first revision of Lewis lesson.

CONCLUSION

Our concept of the role of expert systems in instruction has changed during the first year. Where originally we planned to have the lesson running in real time in LISP on a standard IBM PC, we are now taking a more conservative approach by developing the end lesson in C. The speed of LISP on the IBM PC and the size of the resulting lesson could be prohibitive. LISP as a development tool, with later conversion to a delivery language, does seem feasible. Still, some of the more sophisticated analysis routines will likely remain on a minicomputer running LISP. For this project, we are using an AI language more as a development and analysis tool than a delivery mechanism. In fact, the mere act of creating an expert and/or student model on a computer is a valuable exercise in itself, even if students never touch a computer. Teachers are not always aware of exactly how experts and students really do solve domain-related problems. Constructing a working model provides tremendous insight into that process, with definite ramifications for how a subject area is taught.

Another realization brought about by our experience is the amount of time that must be spent in initial theory and planning. It takes quite a while to map even a relatively simple knowledge domain such as Lewis structures. A great deal of time must be spent in organizing the many components of a complex lesson. The Lewis structure project could have taken an entire semester in planning alone, without ever producing a pilot of the lesson. Related to this is the amount of time consumed by a steep learning curve as new techniques, languages, machines, and theories are applied to hack out a path to a working, worthwhile lesson.

All those who have worked in relatively new environments are aware of the surprises one can expect--equipment failures, lack of equipment, language incompatabilities, staff turnover, and many more. Also, the lack of extensive development tools such as a graphics editor made certain tasks lengthy and tedious. Without some of the existing routines we borrowed and some off-the-shelf symbolic debuggers, the project would not be as close to its current stage of development.

With the background of this and other ICAI projects, and the continued communication among those working in the field, projects such as the Lewis structure lesson should eventually bring to students and teachers ever more powerful ways to increase the efficiency and quality of instruction.

BIBLIOGRAPHY

Sleeman, D. and Brown, J. S., ed. Intelligent Tutoring Systems. (Academic Press: New York, NY, 1982).

Barr, Avron and Feigenbaum, Edward, ed. Handbook of Artificial Intelligence, 3 volumes. (HeurisTech Press: Stanford, CA, 1982).

LEARNING STYLES AND EMBEDDED TRAINING:
A CASE STUDY

Mary L. Stoddard
Los Alamos National Laboratory
INFORM Office, MS B260
Los Alamos, New Mexico 87545
(505) 667-9444

ABSTRACT

The use of online help systems as a form of training is recent, with the emergence of embedded training. Embedded training reduces the need for mode changes. More study of user behavior with embedded training systems is needed for design considerations. This study reports on user behavior with fixed format and context-sensitive online help on a Los Alamos National Laboratory information system and suggests that user learning style is a key determinant for the design of embedded training.

INTRODUCTION

The realization that online help is a form of training for users of interactive computer systems has recently emerged (e.g., Fawcette, 1983; Stoddard, 1984; Heise, 1985; Kearsley, 1985). There are several reasons for this:

- The emergence of easier-to-use menu-driven and iconic interfaces has reduced the dependence on large printed reference manuals.
- The discovery, through user behavior studies, that many people do not learn effectively with printed documentation.
- The high cost of training, both in terms of the user's time investment and institutional costs of training, and management's desire to reduce training costs.
- The increase in the number of remote users of software.
- The advancements in software design resulting in the ability to integrate computer-based training and help with the actual system.
- The observation that mode changes required by separate training or help packages reduce user productivity.

Kearsley (1985) defined embedded training as "on-line instruction which is an integral part of a system or product; instead of isolating learning from what is to be learned, embedded training makes it an ongoing aspect of that system or product." As examples of embedded training he cites helps, simulations, and intelligent tutors, and he fully develops a rationale for why embedded training is the direction to head for in computer-based training. Because of the recency of embedded training (or online help seen as a form of training), few studies have been performed on user behavior with existing embedded training systems. Those that have been performed do not consider user learning style as a critical variable, but rather emphasize degree of prior computer experience only (e.g., Cohill and Williges, 1982; studies reported by Houghton, 1984).

The purpose of this paper is to describe a study of user behavior with fixed-format and context sensitive help, which has been implemented on a personnel information system at Los Alamos National Laboratory, and to suggest that user learning style is a key determinant for the design of embedded training. User behavior was studied for a nine-month period (October 1984-June 1985) through direct observation, consultants' logs, and interview methodologies.

ONLINE HELP AS EMBEDDED TRAINING

At Los Alamos National Laboratory, a Department of Energy contractor with 7600 employees, a multi-functional and menu-driven set of information systems called INFORM has been developed for use by managers and administrators. For the more complex data entry systems, extensive documentation (in the form of walk-through quick guides) and live training in a computer classroom have been offered. Several different formats for training have been used and evaluated, including computer-based training, traditional classroom instruction, and hands-on workshops (see Stoddard, 1985).

One personnel information system, called the Salary Review System, has been in production since December 1984. This system enables Laboratory managers to calculate and propose raises for their employees, to generate reports on employee data for use in their decision making, and to approve raises proposed by subordinate managers. The system was developed in house, is written in COBOL, uses the forms management system (FMS) of Digital Equipment Corporation, and runs on VAX/VMS machines. Extensive online help is provided throughout the system. This online help is accessed in two ways:

1. *Fixed-format help: HE menu option*
   The main menu of the Salary Review System contains 11 options that perform the system functions. One option is HE, which stands for HELP. When HE is typed at the cursor, the users are prompted for the option they want a detailed description of. They then type in a three-letter option code (e.g., RRP for raise review reports, CWA for change write access) and are given an informational screen describing the function and usage of that option (see Figures 1 and 2).

2. *Context-sensitive help: Help key from data entry screens*
   If users press the PF2 key (on a VT100) or the F2 key (on an IBM PC), they receive one line of help information. This line either provides a description of the task at hand or what the next logical choices are. If users press this key twice, they receive a full screen of help information that describes in greater detail the current function and the various ways one can proceed. This help is context sensitive in that several different help messages and screens are available from each work screen; the message or help screen displayed is dependent upon where the user's cursor resides at that point in time on the screen (see Figures 3 and 4 for examples).

Users are made aware of the existence of the online help from screen prompts and the printed documentation. In addition, online help is discussed and demonstrated in the hands-on workshops provided for the system.

THE USERS

Approximately 350 Laboratory employees use the Salary Review System. The majority are managers (275), with the remainder being their administrative assistants and the Laboratory's personnel department staff. Because Los Alamos National Laboratory has an extensive scientific computing network, many of the Laboratory managers have extensive computer literacy on command language systems. In addition, many have recent experience using personal computers as administrative tools. A significant percentage of the population (20%), however, have had either little prior exposure to computers or only experience with electronic mail and word processing systems.

METHODOLOGY

User behavior has been monitored three ways: direct observation during workshops and during actual use, logging of consulting phone calls, and interviews. Information was sought to answer the following questions:

- What type of training most effectively produces the greatest amount of learning for these users? What are the preferred learning styles of these users?
- How frequently do users access online help or perceive they access online help?
- How many consulting calls received could have been answered by online help, and what learning styles do the frequent callers display?
- How can the online help be modified to better meet user needs?
- Was the time invested in generating the extensive online help cost effective?

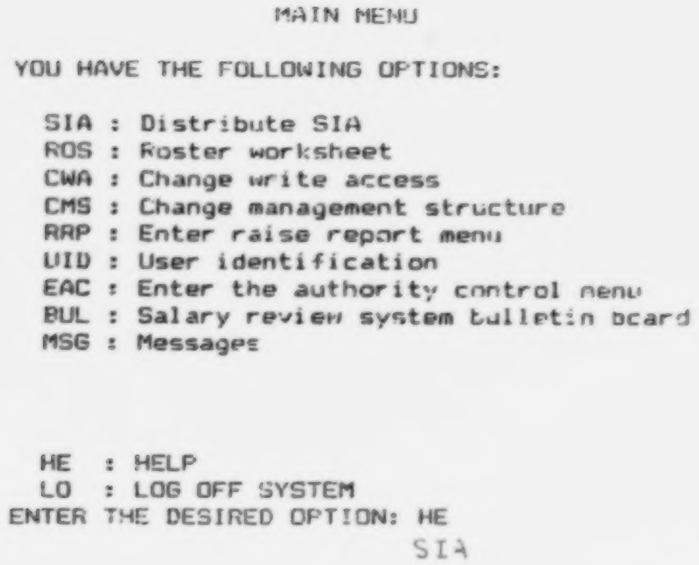
Direct Observation During Workshops: Some students from each workshop (total 50) were observed to see if they spontaneously used the online help when they were in trouble, if they had even been aware of the availability of online help (learned through the verbal, screen, and printed prompts), and if the online help accessed answered the users' questions.

Direct Observation During Actual Use: On occasion, our user services staff has the opportunity to observe users using the system to perform the task of salary review or report generation. Observations here were the same as in workshops, but the setting was more natural for the users and thus a better indication of whether the online help would really be useful.

Consultants' Log: Users may call our consulting office during working hours to obtain answers to questions on data or usage and to report problems. The percentage of problem versus usage questions is noted. For some usage calls, it is noted whether the question could have been answered with the online help.

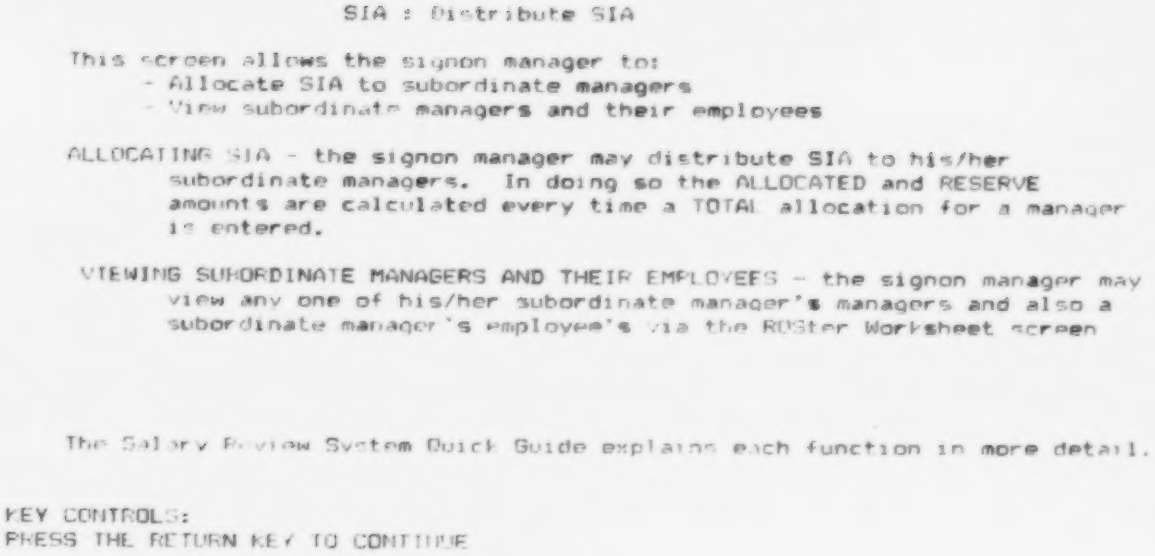
Interviews: Because user satisfaction is a high priority for our information system, the user services staff regularly discussed system design and efficiency with users. During many of these discussions, both in person and by phone, users were asked if they use the online help and if so how often and whether it was helpful.

BEST COPY AVAILABLE



```
                    MAIN MENU

      YOU HAVE THE FOLLOWING OPTIONS:

        SIA : Distribute SIA
        ROS : Roster worksheet
        CWA : Change write access
        CMS : Change management structure
        RRP : Enter raise report menu
        UID : User identification
        EAC : Enter the authority control menu
        BUL : Salary review system bulletin board
        MSG : Messages



        HE  : HELP
        LO  : LOG OFF SYSTEM
     ENTER THE DESIRED OPTION: HE
                                    SIA
```

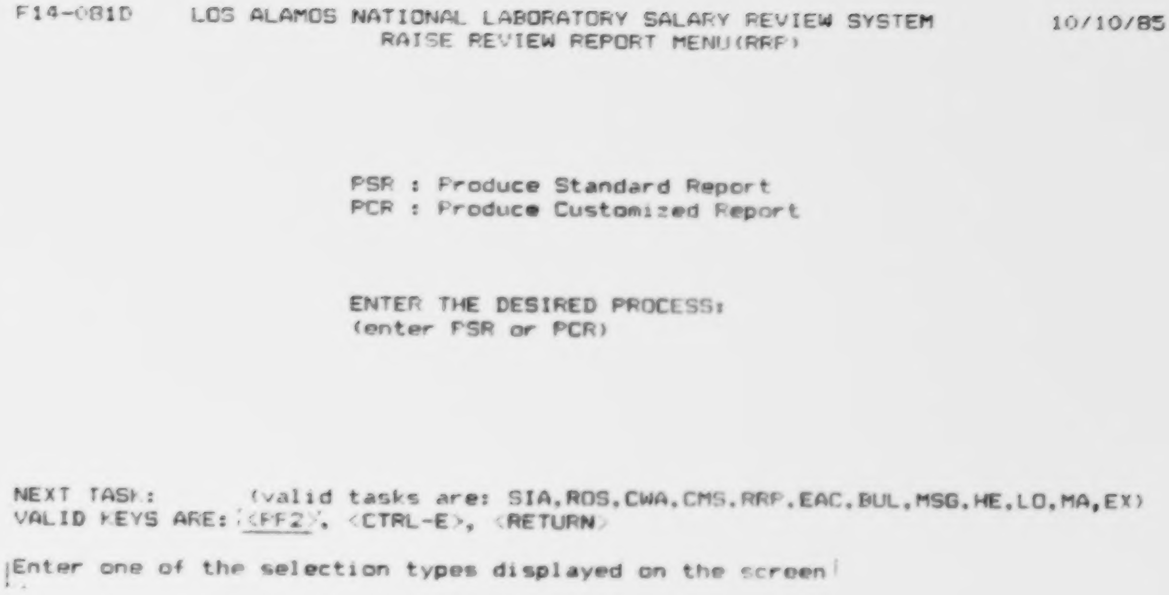
Figure 1: Access to Fixed-Format Help.



```
                      SIA : Distribute SIA

     This screen allows the signon manager to:
           - Allocate SIA to subordinate managers
           - View subordinate managers and their employees

     ALLOCATING SIA - the signon manager may distribute SIA to his/her
             subordinate managers.  In doing so the ALLOCATED and RESERVE
             amounts are calculated every time a TOTAL allocation for a manager
             is entered.

      VIEWING SUBORDINATE MANAGERS AND THEIR EMPLOYEES - the signon manager may
             view any one of his/her subordinate manager's managers and also a
             subordinate manager's employee's via the ROSter Worksheet screen



     The Salary Review System Quick Guide explains each function in more detail.

KEY CONTROLS:
PRESS THE RETURN KEY TO CONTINUE
```

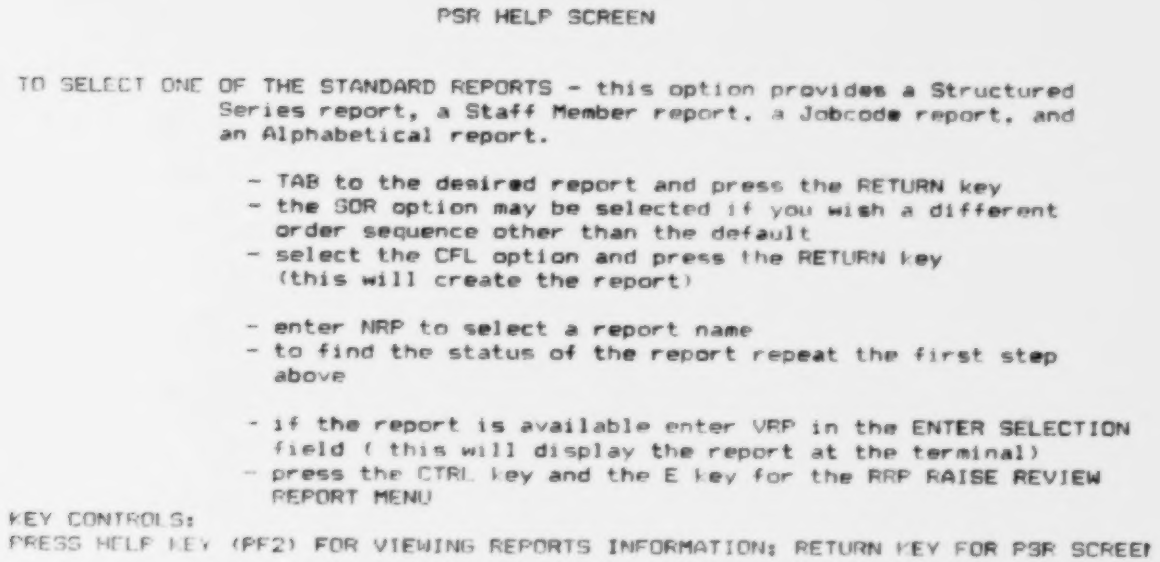
Figure 2: Example of a Fixed-Format Help Screen

BEST COPY AVAILABLE



```
F14-081D    LOS ALAMOS NATIONAL LABORATORY SALARY REVIEW SYSTEM          10/10/85
                         RAISE REVIEW REPORT MENU(RRP)



                    PSR : Produce Standard Report
                    PCR : Produce Customized Report


                    ENTER THE DESIRED PROCESS:
                    (enter PSR or PCR)




NEXT TASK:       (valid tasks are: SIA,ROS,CWA,CMS,RRP,EAC,BUL,MSG,HE,LO,MA,EX)
VALID KEYS ARE: <PF2>, <CTRL-E>, <RETURN>

Enter one of the selection types displayed on the screen
```

Figure 3: Example of One Line of Context-Sensitive Help: One Press of Help Key.



```
                               PSR HELP SCREEN

 TO SELECT ONE OF THE STANDARD REPORTS - this option provides a Structured
               Series report, a Staff Member report, a Jobcode report, and
               an Alphabetical report.

                 - TAB to the desired report and press the RETURN key
                 - the SOR option may be selected if you wish a different
                   order sequence other than the default
                 - select the CFL option and press the RETURN key
                   (this will create the report)

                 - enter NRP to select a report name
                 - to find the status of the report repeat the first step
                   above

                 - if the report is available enter VRP in the ENTER SELECTION
                   field ( this will display the report at the terminal)
                 - press the CTRL key and the E key for the RRP RAISE REVIEW
                   REPORT MENU
KEY CONTROLS:
PRESS HELP KEY (PF2) FOR VIEWING REPORTS INFORMATION; RETURN KEY FOR PSR SCREEN
```

Figure 4: Example of Full Screen of Context Sensitive Help: Two Presses of Help Key.

Measures were all qualitative. An additional quantitative method of monitoring user behavior would have been useful: a software monitor which displays the frequency and context of use of the online help. Unfortunately, this was not done because of the lack of time to develop such a monitor and the overhead of such monitors resulting in a slowing of response time. The priorities of this system were meeting the schedule and providing a new system for managers. Although the lack of quantitative results does detract from the predictability of the results of this study, this study does represent a real-world situation where ongoing qualitative measures can significantly impact future work. The evaluation methodologies used here are commonly used in the fields of human factors in computing and cognitive psychology.

RESULTS

Learning Styles

Observations of users' preferred methods of learning how to use the Salary Review System revealed three preferred styles:

1. DISCOVERY LEARNING STYLE: Some users preferred to learn how to use the system completely on their own, with no formal training. These users log onto the system and browse, showing little computer anxiety. They tried the different menu options to see what happened and used printed documentation as a reference when they could not obtain answers from the screen. Use of the phone consulting service was reserved for questions not answered in documentation. These users tended to be somewhat impatient in their learning, desiring to learn system features only as they needed to use them.

2. GUIDED DISCOVERY LEARNING STYLE: These users preferred to learn by starting to use the system in a structured computer classroom or one-on-one training situation, but after a very short time learning the basics (15-30 minutes) they became impatient and wanted to discover more about the system on their own. They referred to the printed documentation step-by-step and then subsequently used it as a reference.

3. STRUCTURED LEARNING STYLE: These users would not begin using the system without formal classroom training. During the training they followed instructor directions carefully and referred to printed documentation when they were asked to do so. They discovered little about the system on their own, so thorough training on all system features was necessary. They often had difficulty finding the information they needed in the documentation and relied heavily on phone consulting after formal training.

User preferred learning style was determined through interviews and direct observation, therefore the categorization is subjectively based.

Frequency of Access of Online Help

Observations revealed a relationship between learning style and frequency of access of online help (see Table 1). Users with a guided discovery learning style seemed more willing to access online help than those with a discovery learning style. In turn, the structured learning style users seemed to access the online help the least.

The guided discovery learning style users learned of the availability of the online help through trainer announcements and documentation, and subsequently used it in addition to printed documentation. Interviews revealed that the discovery learning style users frequency of access of online help seemed dependent upon their prior preference of printed versus online documentation. A study performed at Los Alamos on a different help system suggested that users who were previously not familiar with online help tended to rely heavily on printed documentation, if they could readily locate it, whereas those who had previously used online help systems successfully tended to use online help more than printed documents (Stoddard et al., 1985). This same pattern could also be operating here, but further study is necessary to verify this hypothesis.

The structured learning style users revealed some computer anxiety, so they seemed reluctant to press any function keys for fear of making an unrecoverable error. This tendency seemed to disappear as the user became more comfortable with the system and if, on a one-on-one training basis, online help was frequently accessed and proven useful.

Context-Sensitive Versus Fixed-Format Help Frequency of Use

Of the two types of help available, the context-sensitive help seemed to be most accessed and most useful to users. The fixed-format help provided very general information on functionality of basic system options from the main menu. The method of accessing it is somewhat awkward in that one cannot browse sequentially through the help screens for all the options, but rather must separately request help (HE) and then the option name. The same information is easily browsed in the printed documentation. Discovery learning style users tended to find out about the function of options by entering the option itself to see what happened, whereas the guided discovery and structured users relied on description from the training or the printed document.

TABLE 1: Relationship of Learning Style to Help Accessed

Learning Style	Job Category	Frequency of HELP Access	Phone Consulting	Preferred Training	Printed Document
Discovers	more technical managers	moderately high	low, only for non-documented problems	self-study	used as a reference
Guided Discovery	more non-technical and technically oriented administrative assistants	high	moderate	formal, followed by self-study	for initial learning and as reference
Structured	more administrative assistants than managers	low	high	formal, followed by phone consulting	as a reference only if guide found useful during class

Context-sensitive help tended to be accessed when users were in trouble and did not know what to do next. Once it was accessed and was found to be useful for troubleshooting, users tended to remember to use it subsequently. Users who became accustomed to using the system without using the help key, however, tended to continue this pattern and not use it. The feature of obtaining an entire screen full of help upon pressing the help key twice was less frequently used. User interviews revealed that, in spite of its being documented, it seemed to be difficult to remember and illogical to press the help key twice when in trouble. Users had not seen this feature in other systems they had used, but expressed appreciation for being reminded about it in the interview. Thus, in spite of the fact that the users interviewed had been told of this feature during formal training, they rarely remembered it during independent use.

SUMMARY AND IMPLICATIONS

For the purposes of answering our initial questions, the results of this study suggest the following:

- Fixed-format help accessed from the main menu is less effective for user learning than context-sensitive help. It could be made more useful by making it easier to browse; that is, not require the user to type in each menu option when asking for HELP and offer all the help screens in succession (with the ability to return to the main menu at any time).
- The feature of the double press of the help key for obtaining more in-depth help was unnatural, although the full screen of help was judged by users as useful once they remembered how to get it. The interface should be modified, therefore, so that the full screen of help will be provided instead of one line of help when the task at hand is sufficiently complex.
- Printed documentation should be modified to include actual examples of help screens and messages.
- Formal training should include more frequent demonstrations of the online help during the hands-on session.
- When training is advertised for the system, the extensive online help feature of this system (combined with printed documentation) should be announced as well as the hands-on workshops to respond to the many users with discovery learning styles.

Several hypotheses can be formulated here that relate to the use of online help as a form of embedded training:

1. The frequency of access of online help is directly proportional to the degree to which the online help is proven useful in demonstrations, training, printed documentation, and prior use of online help.

2. The degree of usefulness of online help for learning a new system is directly related to the degree of naturalness of access to the help.

3. Learning style is an important variable when designing embedded training. Learning style is dependent not only on prior computer experience but also on a user's job and prior experience with online and printed computer documentation. Since it is experience-based, however, this variable must be viewed as dynamic within an individual.

4. If a system is well documented and has an easy to use interface, and if the user population is dominated by users with discovery or guided discovery learning styles, context-sensitive help may replace the need for formal training or more complex, structured computer-based training packages.

These hypotheses are useful on several counts. We can use them in our context for future planning of online help and for screening users for formal training. They are useful in other contexts because they suggest that the specific design of embedded training depends upon the dominant learning style of the user population. Thus, before embarking on a project to develop embedded training, project developers need to include criteria not normally examined in an audience analysis. These hypotheses are also useful to the research community in that they provide several specific areas where experimental studies could have a significant impact on the direction of embedded training. Our case study also demonstrates that commonly used evaluation methodologies are effective in collecting user behavior information for systems already in production.

The degree of needed sophistication of embedded training is a critical decision for the cost-benefit analysis in system development. As both Walker (1985) and Kearsley and Seidel (1985) point out, highly user responsive context-sensitive help requires a large effort, and instructional designer involvement at an early stage is required. Our case study demonstrated, however, that help which is context sensitive (but not "intelligent") does not require major increases in software development resources and yet it does seem to have a positive impact on ease-of-learning and ease-of-use. The recognition that the development of online help can be cost effective if integrated into other phases of the software life cycle is not new (e.g., Relles et al., 1981), and yet demonstration that embedded training will further enhance cost effectiveness is not fully recognized.

ACKNOWLEDGMENTS

This study was performed under the auspices of the Department of Energy. The software for the Salary Review System was developed by the Administrative Data Processing Division, with Hal Armstrong being the project leader. The help screens themselves were developed by Annette Fernandez. The INFORM Office is led by Joe Thompson; direct user consulting is performed by Kay Fletcher and Deanna Brooks.

REFERENCES

Cohill, A. and R. Williges. 1982. "Computer-augmented retrieval of help information for novice users." *Proceedings of the Human Factors Society*, pp. 79-82.

Fawcette, J. 1983. "Finally, some help for documentation." *Personal Computing*, December, pp. 141-145.

Heise, H. 1985. "Embedded Training." Handout from *Data Training* Conference session.

Houghton, R. 1984. "Online help systems: A conspectus." *Communications of the ACM* 27 (2), pp. 126-133.

Kearsley, G. 1985. "Helps, simulations, and intelligent tutors: An overview of embedded training." *Data Training*, May, pp. 32-38.

Kearsley, G. and R. Seidel. 1985. "Automation in training and education." *Human Factors* 27 (1), pp. 61-74.

Relles, N., N. Sondheimer, and G. Ingargiola. 1981. "A unified approach to online assistance." *Proceedings of the National Computer Conference*, pp. 383-388.

Stoddard, M. 1984. "Automated user assistance." *Data Training*, September, pp. 36-38.

Stoddard, M. 1985. "Critical variables in start-up training for multi-functional information management systems." *Proceedings of the Human Factors Society Conference*.

Stoddard, M., K. Berkbigler, E. Peter, and R. Wheat. 1985. "User behavior upon introduction of a network help system." *SIGCHI Bulletin*, January, pp. 25-41.

Walker, J. 1985. "Implementing documentation and help online." ACM CHI '85 Conference Tutorial Handout.

A MODEL FOR RESEARCH, DESIGN, AND SELECTION OF INSTRUCTIONAL USES OF TECHNOLOGY IN MATHEMATICS AND SCIENCE EDUCATION

Arthur L. White - The Ohio State University-Columbus

Donna F. Berlin - The Ohio State University-Newark

ABSTRACT

This paper presents a theoretical, instructional model based upon current learning theory, prevailing instructional strategies, and technological developments. Instructional activities based upon the intersection of two continuums are described. The two continuums are representational level (i.e., concrete, semiconcrete, abstract) and movement (i.e., manipulation, animation, static). A collection of instructional strategies and activities appropriate to mathematics and/or science education which employ technological tools (e.g., calculator, computer) have been selected. These activities are discussed in terms of their relationship to the instructional model, learning theory, and empirical evidence. A research agenda is outlined for exploring and evaluating the theoretical model in terms of technology, objectives of the curriculum areas, and learner needs and preferences.

Education (learning and instruction) is a rational enterprise which can be studied and continually made more effective and efficient. The study of learning involves the interrelationships and interactions between the learners, the learnings (that which is to be learned), and the learning environments (see Figure 1).

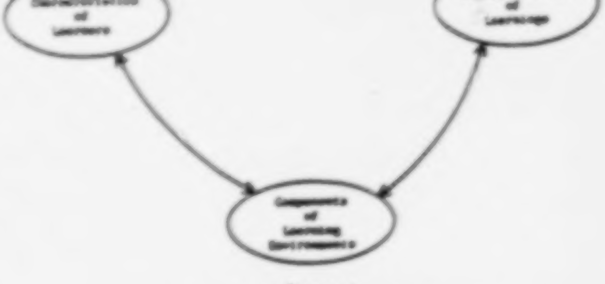


Figure 1
Principal Factors of Learning Context

This paper will present a theoretical model based upon current learning theory, prevailing instructional strategies, and technological developments applied to mathematics and science education. Instructional activities based upon the intersection of two continuums will be described. The two continuums are representational level (i.e., concrete, semiconcrete, abstract) and movement (i.e., manipulation, animation, static). A research agenda to explore and evaluate this theoretical model will also be suggested.

The proposed theoretical model focuses upon the components of the learning environments which include interactions among teaching methods, teaching materials, teaching techniques, delivery systems, presentation modes, motivation techniques, environmental features, and environmental organization. The incorporation of rational thought, theory, and empirical evidence is critical in the design of instructional systems within the learning environments. A model or paradigm can then be developed to guide in the design of instructional systems and provide a means to investigate specific, delineated dimensions of the instructional systems. Figure 2 provides examples of dimensions that could be used for instructional model development.

Concrete----------Semiconcrete----------Abstract
Expository---------Guided Discovery------Discovery
Rote Nonsense------Rote Relevant---------Meaningful
Passive------------Reactive--------------Proactive
Manipulated--------Animated--------------Static
Unidimensional-----Bidimensional---------Multidimensional
Random-------------Random Within Fixed Domain---------Fixed
Reception----------Reception Within Fixed Domain------Selection
Teacher Determined------------Teacher/Learner Determined-------Learner Determined
Unimodal-----------Bimodal---------------Multimodal
Single Embodiment------------Biembodiment----------Multiembodiment

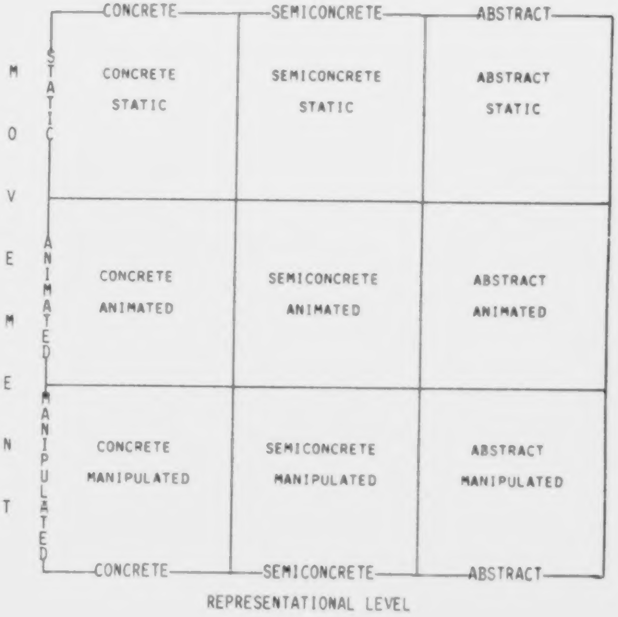
Figure 2. Dimensions for Instructional Model Development

Learning is defined as a change in the individual's cognitive structure, one's previously acquired storage of information. It is assumed that learners interact actively with the environment (interactionalism) and construct meaning from their experiences (constructivism). The work of Ausubel, Bruner, Dienes, and Piaget is representative of this perspective.

Educators have generally accepted a model of concept development based upon a progression from concrete to semiconcrete to abstract representation. Piaget's stages of development (sensorimotor, preoperational, concrete operational, and formal operational) and Bruner's stages to represent the world (enactive, iconic, and symbolic) support this view. This has been translated into instructional practice as manipulation of concrete objects, pictorial or graphic representation, and finally symbolic form. This sequence describes the representational continuum. Salomon (1974) suggests that representational level is a basis for describing and differentiating forms of instructional media.

Wittrock's generative model of learning suggests that learners generate associations between stimuli and prior learning and that both prior concrete experiences and verbal abilities are used to construct these associations. Instruction that involves different methods for different learners in different situations facilitates the generative process.

The proposed instructional model attends to the dimensions of representational level (concrete--semiconcrete--abstract) and movement (manipulated--animated--static) as illustrated in Figure 3.



MOVEMENT	CONCRETE	SEMICONCRETE	ABSTRACT
STATIC	CONCRETE STATIC	SEMICONCRETE STATIC	ABSTRACT STATIC
ANIMATED	CONCRETE ANIMATED	SEMICONCRETE ANIMATED	ABSTRACT ANIMATED
MANIPULATED	CONCRETE MANIPULATED	SEMICONCRETE MANIPULATED	ABSTRACT MANIPULATED
	CONCRETE	SEMICONCRETE	ABSTRACT

Figure 3

Instructional Model

These two dimensions were selected because they are consistent with the learning theory and reflect instrucional practices valued in the teaching of mathematics and science: activity-based learning, hands-on activities, multimedia environments, and learner-centered. Of particular interest is the role of technology in the development and modification of the instructional model and the use of the technology in the design and selection of the instructional materials and activities.

Dimension I, Representational Level, describes a continuum from concrete to abstract representation. The concrete representational level is simply interpreted as a learning activity which involves concrete objects which represent physical entities. The semiconcrete level involves representation of the objects in the form of pictures, cartoons, or images as displayed by media such as television, computer monitors, overhead projectors, filmstrips, films. The abstract representational level engages the learner with ideas, verbal descriptions, and symbolic representation of the concrete or semiconcrete experiences. Mathematical symbols and formulas are examples of the abstract level.

Dimension II, Movement, is a continuum from manipulated to static form. It is fundamentally related to two factors: change and the control of that change. In the static condition there is no change and evidence of change can only be inferred, not observed directly. Animation involves an observable change but the control of that change and the factors influencing that change are not controlled by the learner. The change can be controlled by the teacher or by some other means such as a computer program. Manipulation is a condition involving observable change in which the learner actively interacts with the learning environment and therefore has a degree of control over the change.

The interaction among the levels of Dimension I, Representation, and Dimension II, Movement, are illustrated by the nine individual cells in the instructional model (see Figure 3).

This instructional model can be applied to various curriculum areas and topics. Figure 4 illustrates a mathematics instructional model for the concept of patterns.

CONCEPT - PATTERN

MOVEMENT	CONCRETE	SEMICONCRETE	ABSTRACT
STATIC	LEARNER OBSERVES PATTERN OF ATTRIBUTE BLOCKS AND CHOOSES THE CORRECT EXTENSION	LEARNER OBSERVES PATTERN ON TASK CARDS AND MATCHES OR COMPLETES THE PATTERN WITH CRAYON	LEARNER OBSERVES SEQUENCE OF NUMBERS AND FINDS THE RULE AND NEXT NUMBER
ANIMATED	LEARNER OBSERVES THE FORMATION OF A PATTERN WITH ATTRIBUTE BLOCKS AND IDENTIFIES THE EXTENSION	LEARNER OBSERVES A COMPUTER SIMULATION OF THE FORMATION OF A PATTERN AND IDENTIFIES THE EXTENSION	LEARNER OBSERVES THE FORMATION OF A SEQUENCE OF NUMBERS ON THE COMPUTER AND IDENTIFIES THE RULE AND THE NEXT NUMBER
MANIPULATED	LEARNER DUPLICATES A PATTERN AND FORMS AN EXTENSION USING ATTRIBUTE BLOCKS	LEARNER DUPLICATES A PATTERN AND FORMS AN EXTENSION USING A COMPUTER SIMULATION OF ATTRIBUTE BLOCKS	LEARNER USES THE CALCULATOR TO GENERATE PATTERNS AND SEQUENCES OF NUMBERS AND TO DETERMINE THE RULE FOR THE SEQUENCE

REPRESENTATIONAL LEVEL

FIGURE 4

INSTRUCTIONAL MODEL - MATHEMATICS

An instructional model for science and the concept of buoyancy is described in Figure 5.

CONCEPT - BUOYANCY

MOVEMENT	CONCRETE	SEMICONCRETE	ABSTRACT
STATIC	LEARNER OBSERVES END STATE OF 'LIQUID DISPLACED BY OBJECT' EXPERIMENT	LEARNER OBSERVES COMPUTER SIMULATION OR PICTURE OF END STATE OF 'LIQUID DISPLACED BY OBJECT' EXPERIMENT	LEARNER OBSERVES TABLE OR GRAPH GIVING THE WEIGHT OF LIQUID DISPLACED AND THE WEIGHT OF THE OBJECT
ANIMATED	LEARNER OBSERVES A DEMONSTRATION OF THE 'LIQUID DISPLACED BY OBJECT' EXPERIMENT	LEARNER OBSERVES A COMPUTER SIMULATION OF THE 'LIQUID DISPLACED BY OBJECT' EXPERIMENT	LEARNER OBSERVES THE CHANGING VALUES OF THE VOLUME/WEIGHT OF LIQUID DISPLACED BY AN OBJECT AND THE HEIGHT OF THE LIQUID ON THE OBJECT
MANIPULATED	LEARNER DETERMINES THE WEIGHT OF LIQUID DISPLACED BY OBJECTS PLACED IN A LIQUID	LEARNER INTERACTS WITH A COMPUTER SIMULATION TO DETERMINE THE WEIGHT OF LIQUID DISPLACED BY OBJECTS PLACED IN A LIQUID	LEARNER CHANGES THE VALUES IN A MATHEMATICAL MODEL COMPARING THE WEIGHT OF OBJECTS AND THE WEIGHT OF EQUAL VOLUMES OF A LIQUID

REPRESENTATIONAL LEVEL

FIGURE 5

INSTRUCTIONAL MODEL - SCIENCE

A final example of the instructional model which integrates mathematics and science through the concept of area is provided in Figure 6.

CONCEPT - AREA

MOVEMENT	CONCRETE	SEMICONCRETE	ABSTRACT
STATIC	LEARNER OBSERVES A RECTANGULAR FRAME FITTED WITH COLOR CUBES	LEARNER OBSERVES A COMPUTER GRAPHIC DISPLAY OF COLORED SQUARES FITTED TO RECTANGULAR FRAMES	LEARNER OBSERVES A TABLE OF DIMENSIONS OF RECTANGLES AND RESULTING AREAS
ANIMATED	LEARNER OBSERVES A DEMONSTRATION OF FITTING COLOR CUBES INTO RECTANGULAR FRAMES	LEARNER OBSERVES COMPUTER SIMULATION OF FITTING COLORED SQUARES INTO RECTANGULAR FRAMES	LEARNER OBSERVES THE FORMATION OF A SEQUENCE OF NUMBERS REPRESENTING THE DIMENSIONS AND AREA OF RECTANGLES
MANIPULATED	LEARNER FITS COLOR CUBES INTO RECTANGULAR FRAMES	LEARNER INTERACTS WITH A COMPUTER SIMULATION TO FIT COLORED SQUARES INTO RECTANGULAR FRAMES	LEARNER INTERACTS WITH THE COMPUTER PROGRAM TO CHANGE THE NUMBERS REPRESENTING DIMENSIONS OF A RECTANGLE AND THE COMPUTER DISPLAYS RESULTING AREA

REPRESENTATIONAL LEVEL

FIGURE 6

INSTRUCTIONAL MODEL - MATHEMATICS/SCIENCE

Current theoretical and empirical literature would suggest an instructional delivery or sequence of these cells in either a horizontal (left to right), vertical (lower to upper), or diagonal (lower, left to upper, right) direction. However, these sequences of activities are in need of additional investigation in terms of the needs and preferences of individual learners and the objectives and requirements of the curriculum areas.

Support for many of the specific activities outlined in the three instructional models specific to curriculum areas can be found in the literature. This literature base will be discussed in terms of the movement levels: manipulated, animated, and static. The use of a variety of manipulative materials and activities has been suggested by Dienes (1971) in his Multiple Embodiment and Mathematical Variability Principles in order to facilitate concept abstraction and generalization. Manipulation of concrete objects is also recommended by Reys (1971) and Suydam and Dessart (1976). The role of visual imagery as advanced by the work of Levin (1976) and Paivio (1971) provides a basis for the activities involving animation. Paivio suggests a dual processing hypothesis, the inherent ability to process information by verbal and visual procedures. Additional support for these activities is found in a research review by Suydam and Higgins (1977) which indicates that teacher demonstration can be as effective as student manipulation. Lastly, activities involving static representations are supported by the work of Ausubel (1968). He suggests that reception learning, material presented in final form, can be meaningful to the learner.

A review of the literature involving the instructional use of technology provides support for some of the other activities described in the cells of the model. There has been nearly a decade of research involving the use of the calculator in the mathematics curriculum. The following comment best summarizes the research findings regarding calculator use:

> The research evidence supports the use of calculators at all grade levels, indicating that achievement will not be harmed and may be enhanced when calculators are used. Furthermore, the evidence indicates that some mathematics content is taught better when calculators are used than when they are ignored. (Suydam, 1982, p. 10)

Support for the instructional uses of the computer have not benefitted from such a comprehensive, long-term period of investigation. Generally, the research has confirmed positive effects associated with the use of computers in education. Barcey (1982) found that students generally learn more, learn in less time, and retain more using computers, and that affective and motivational effects are positive. Bork (1979) has championed the use of computer simulations, analogs of real events, replete with graphics, animation, manipulation, interaction, and three-dimensionality. Forman (1985) has hypothesized a new representational level resulting from the instructional use of computers. His term "kinetic print" refers to the interaction with active iconic symbols. Clearly, the role of technology in an instructional model should be considered and investigated.

The following agenda for research is suggested:

1. How does the variation and sequence of activities along the representational levels continuum influence the learner's cognitive

structure, information processing, and understanding of mathematics and science concepts and processes?

2. How does the variation and sequence of activities along the movement dimension continuum influence the learner's cognitive structure, information processing, and understanding of mathematics and science concepts and processes?
3. Do the influences of representational level and movement in clinical instructional systems generalize to the complex learning environments of the classroom?
4. Can the representational level and movement dimensions be manipulated in instructional systems to match learner preferences and to provide transitional experiences toward development of cognitive restructuring of information and abstract thinking?
5. Can educational technology be integrated with other modes to manipulate the representational level and the movement dimension in order to facilitate the learner's development of knowledge structures, thinking processes, and application skills for mathematics and science concepts?
6. Can a sequence of activities which varies in levels of representation and movement be identified to help correct misunderstandings, partial understandings, and confusions about mathematics and science concepts?

These are a few of the questions that need to be answered so that instructional systems can be carefully and systematically designed in order to maximize the benefits of diverse instructional tools and promote equitable and efficient learning.

References

Ausubel, D.P. (1968). *Educational psychology: A cognitive view*. New York: Holt, Rinehart & Winston.

Bork, A. (October, 1979). Learning with computer simulation. *Computer*, 76-84.

Barcey, G.W. (November/December, 1982). Computers in education. What the research shows. *Electronic Learning*, 51-54.

Bruner, J.S. (1960). *The process of education*. New York: Vintage Books.

Dienes, Z.P. (1971). *Approach to modern mathematics*. London: Hutchinson Educational.

Forman, G. (1985). The value of the kinetic print in computer graphics for young children. In E. Klein (Ed.), *Children and computers. New directions for child development*, (No. 28, pp. 14-35). San Francisco: Jossey-Bass.

Levin, J.R. (1976). What have we learned about maximizing what children learn? In J.R. Levin and V.L. Allen (Eds.), *Cognitive learning in children: Theories and strategies*. New York: Academic Press.

Paivio, A. (1971). *Imagery and verbal processes*. New York: Holt, Rinehart, & Winston.

Piaget, J., & Inhelder, B. (1969). *The psychology of the child*. New York: Basic Books.

Reys, R.E. (1971). Considerations for teachers using manipulative materials. *Arithmetic Teacher*, *18*, 551-558.

Salomon, G. (1985). The value of kinetic print in computer graphics for young children. In E.L. Klein (Ed.), *Children and computers. New directions for child development*, (No. 28, pp. 19-35). San Francisco: Jossey-Bass.

Suydam, M.N. (1982). *The use of calculators in pre-college education: Fifth annual state-of-the-art review*. Columbus, Ohio: Calculator Information Center.

Suydam, M.N. & Dessart, D. (1976). *Classroom ideas from research on computational skills*. Reston, Virginia: National Council of Teachers of Mathematics.

Suydam, M.N. & Higgins, J.L. (1977). *Activity-based learning in elementary school mathematics: Recommendations from research*. Reston, Virginia: National Council of Teachers of Mathematics.

Wittrock, M.C. (1978). The cognitive movement in instruction. *Educational Psychologist*, *13*, 15-29.

SEVEN GUIDELINES FOR CAL PROJECT TIME MANAGEMENT

Raymond Ng

MICROTEL Learning Services

One primary responsibility of a project manager is to produce a quality product within a pre-determined time frame. In CAL development, this task is often complicated by continual introduction of new techniques, unexpected revisions and last minute client input. The following guidelines provide a partial solution to this problem:

1. Limit the production time of any single project to less than six months. New technology can make earlier progress in a long project obsolete. If necessary, split a project.

2. Adopt one project life cycle across the department to ensure uniformity of procedures. Uniformity increases efficiency.

3. Involve everyone in the team to plan production objectives, time-lines and task assignments.

4. Record the actual time in completing a task. Use previous records to predict future time-lines.

5. Plan the production schedule so that design, development and evaluation of different parts of the project may take place simultaneously.

6. Involve the subject matter expert, management and the client very early in the production.

7. Prepare for the unexpected. Allow ample buffer time in the production schedule. Be tactically flexible but strategically rigid.

No guideline can apply to all cases. When to apply these guidelines depends on the individual production environment. How to apply them is a virture to be gained only from experience.

THE JOINT STAFF OFFICER TRAINING SYSTEM (JSOTS)

Raye Newmen

Navy Personnel Research and Development Center

The objective of the JSOTS project is to prepare Action Officers (O-5 officers from all four branches of service) assigned to the Organization of the Joint Chiefs of Staff to quickly and competently assume their duties, which involve a wide range of defense policy, planning and operational issues. The job tasks of AOs include primarily higher cognitive areas such as decision making and discernment based on a joint military perspective, and complex skills such as effective negotiation. Even the less "fuzzy" and better defined training areas such as the organizational structure and administrative procedures of OJCS tend to be quite complex.

This presentation will describe the process followed to analyze these often "soft" training objectives and build computer-based instructional units based on those analyses. We will describe our experiences using the Computer-Based Educational Software System (CBESS) as the principal authoring system for the project, and point out the hardware and personnel requirements for instructional systems development and delivery using CBESS.

The implications of the JSOTS project for developing training for high level professionals in other settings will be discussed. Process evaluation data will be presented as an indication of system effectiveness. A demonstration of a JSOTS CBI training unit using interactive video disk will be conducted.

AN INSTRUCTION SYSTEM USING
DATA BASE MANAGEMENT SOFTWARE

James D. Jensen

National Technical Institute for the Deaf

For some instructional situations, data base management system may be useful for both interactive instruction and the associated record keeping. Often, instructional programs are heavily dependent on data files. A data base management program having good programming capabilities may be a good choice for developing CAI, interactive testing and record keeping in such a situation. In addition to vastly simplifying the file maintenance routines, the programs can conveniently be made to automatically update student records and provide progress reports to both students and teacher.

This presentation illustrates the concept with a simple system and discusses considerations in electing to use a DBMS as an alternative to other kinds of software. Handouts include sample files and programs written in dBASE-II.

APPROACHES TO COMPUTER SIMULATIONS FOR MANAGEMENT TRAINING

Stephen Greenfield, Gary Moehnke

Control Data Corporation

Simulations are among the most popular and effective modes of CBT. They are most successful and widespread in domains that lend themselves to quantitative analysis, such as engineering, medicine, or finance. However, many areas with critical training needs, such as management, organizational development, and interpersonal dynamics are not as easily quantified, and are, therefore, not as easy to model for purposes of a dynamic simulation.

It is important to distinguish instructional simulations based on mathematical models, which respond dynamically in complex ways to changes in input variables, from simpler "situation simulations" where situations are presented, choices are made, and corresponding outcomes are shown. The former are potentially the most realistic and flexible, but are also the ones that require specific statement of rules and that are difficult in qualitative areas.

Several examples illustrate the issues involved and potential approaches to augmenting management experience through simulations:

1. A team communications simulation where a complex situation develops and types of communication must be chosen.
2. A motivational dynamics course where management actions determine the outcome of a game-like business simulation.
3. A time management simulation, a quantitative topic.
4. A performance management simulation, where choices in setting expectations are analyzed along multiple dimensions and used for feedback, including effect on a simulated business.

USING DATA BASE MANAGEMENT SYSTEM SOFTWARE ON PERSONAL COMPUTERS FOR INSTRUCTION

Louis E. Frenzel, Jr.

Teknowledgy Sources Inc.

Standard DBMS software such as Ashton Tate's dBASE II/dBASE III, Microrim's RBase 4000, and others can be used as a type of authoring system for creating CBI. The resulting programs are a unique kind of CBI that does not compete with other methods.

The data base or inquiry method of CBI has never been popular or widely known. However, the wide availability of personal computers and low cost DBMS software makes this form of CBI a truly cost and time efficient alternative in some training and productivity improvement situations. DBMS software, like an authoring system, greatly speeds up, simplifies and lowers the cost of development.

To create CBI using a DBMS, you divide the material to be learned into very short "chunks" suitable for entering as a record in the DBMS. One field in each record contains references to other related records or "chunks". An example is a glossary with each definition being entered as a record.

To use the system, the student simply wanders around randomly and repetitively accessing different definitions or "chunks" of interest. The reference pointers suggest other records to access. A complete sequence of terms forms a brief overview of a particular topic.

This approach implements a Piaget-like learning system where the student learns by the discovery method. It is fast, efficient and gets right to the point. Unlike in some traditional CBI, the student is in full control and has random access to the knowledge. He or she learns exactly what they need to know and nothing more. The system is also an excellent on-line job aid and a simpler alternative to an artificial intelligence expert system in some applications.

This type of CBI can be made even more student friendly and effective when one of the new artificial intelligence natural language front-ends is used on the DBMS. Examples are Excalibur's SAVVY and Microrim's RBase 4000/Clout.

HOW ARTIFICIAL INTELLIGENCE TECHNIQUES CAN BE USED IN DEVELOPING COURSEWARE

Brian L. Dear

Hazeltine Corporation

Until recently, the use of artificial intelligence (AI) as a potential tool to aid in the development of computer-assisted instruction (CAI) was considered unrealistic. However, technology has now reached the point where large memory capacities for the storage of information are available for not only mainframe systems, but, more importantly, personal computers. It is now possible to seriously consider using AI methods in the courseware authoring process.

CAI's primary focus is that of the human as learner. AI emphasizes the computer as learner. While each discipline has begun to collect a significant body of literature, little research has been shared, in spite of this relationship to "learning", between the two fields since the two began some twenty-five years ago. The techniques that have arisen out of research in AI can offer a new approach for achieving the full potential of the computer in the development of CAI courseware.

Courseware authoring has traditionally been accomplished through a free-form method of specialized programming language, through menu-driven authoring systems, or through a combination of both. Although the potential already exists in many of the available authoring systems and languages to make use of AI applications, few facilities have been actually implemented. This presentation shall examine several different approaches to incorporating the techniques and methods of AI into CAI. Such techniques as knowledge-based systems design, expert systems, and heuristics have the potential to greatly enhance the authoring process. For example, the potential exists for a fully implemented on-line instructional systems development process in a comprehensive computer-aided authoring (CAA) system. By using an "intelligent" CAA system with as high a level of interaction as that of CAI courseware, this potential can be realized.

THE PROJECT TO EVALUATE AUTHORING SYSTEMS:

A PROGRESS REPORT

Victoria L. Clegg, Jane M. Lommel

Kansas State University

The project to evaluate authoring systems has been undertaken by Kansas State University's Office of Educational Improvement in response to the profusion of micro-computers in education and industry, and the burgeoning interest in computer-based learning.

The project is directed by Victoria L. Clegg and Jane M. Lommel. Their presentation will be the first six month progress report for the project. They will focus on the development of comprehensive evaluation criteria, which is the first step in conducting a criteria based evaluation of micro-computer based authoring systems.

The first phase of the project is to develop fair and comprehensive evaluation criteria that will be valuable guides to educators and trainers in selecting which system(s) to use. These criteria, established with the cooperation of CBT experts and authoring system developers, will describe and weight the importance of authoring system attributes.

In the second phase of the project, commercially available authoring systems that have been contributed to the project will be evaluated according to the criteria established in the first phase.

Then a comprehensive project report will be prepared that reports the results of the quantitative and qualitative evaluations of the authoring systems. The research methodology will also be presented in detail.

USING COMPUTERS IN COUNSELOR TRAINING

William E. Schulz

University of Manitoba

A one-hour, computer program was developed to assist career counselor trainees in developing their goal-setting and initial interview skills. The computer program assists trainees to learn and practice the following skills and tasks:

1. Develop a positive atmosphere in which the counselor is perceived as willing to help.

2. Help client state a specific concern.

3. Elaborate the concern to discover constraints.

4. Obtain clarity as to the purposes of the interview.

5. Obtain greater clarity as to the actual counselee constraints.

6. Evaluate strengths and barriers in relation to the constraint.

7. Examine possible unspoken assumptions which color the constraint.

8. Change the concern/constraint statements to a contingency statement.

9. Evaluate the contingency statement.

10. State a specific counseling goal or action plan.

Participants will have an opportunity to examine printed copies of this computer-assisted, counselor training program for goal-setting.

COMPUTER BASED PREVENTIVE MAINTENANCE TRAINING FOR THE BUS INDUSTRY

Allan Williams

Computer Based Training Services

Computer based training (CBT) programs for technical audiences often require students to read and interpret instructions or questions from the computer monitor. However this requirement often becomes a barrier to learning and reduces the effectiveness of the training package.

This presentation focuses on the use of a combination of both interactive audio and video in preventive maintenance training of bus service personnel. Mechanics typically prefer to look and listen rather than read and thus tend to be prime candidates for the CBT system described. The actual program was specifically related to the maintenance and repair of bus bodies, an area not well understood by the target group.

The system consists of a stereo casette recorder, a video tape recorder (VCR), and a micro computer. The audio component requests input and provides feedback to the user. The video component reinforces the learning by showing the relevant frame to the student. Text on the screen is kept to a minimum and functions primarily as a backup to the audiotape.

Attention is given to design problems, considerations made in developing the package and user problems. The system will be demonstrated and comments offered on the further application of the technology to other technical training areas.

THE EVOLUTION OF A COMPUTER LAB: TOOL OF TECHNOLOGY TRANSFER IN INTERNATIONAL EDUCATION

Claire Bradin, Larry Cross

Economics Institute

The Institute is a transitional training program for international students. Its objective is to prepare them to enter graduate programs in economics and business. The curriculum includes instruction in English as well as economics, accounting, mathematics, statistics, management, finance, and computer science.

In 1983, in recognition of the importance of computer literacy to all students entering American universities, a small instructional computer lab was first established. The rapid growth of the facility may be outlined in the following stages:

1. Awareness - faculty education.

2. Experimentation - Radio Shack computers used for word processing and simple drill-and practice programs.

3. Commitment - needs identification, search for software, purchase of Apple computers.

4. Expansion - purchase of IBM AT's with local area network, use of telecommunications.

5. Present stage - CBI incorporated into the curriculum in many areas.

6. Future plans - software development.

Each of these stages will be discussed in detail in the presentation. Handouts giving specific information about the software selected will be provided.

In less than two years, the computer lab has become an integral part of the instructional program. It has been found that foreign students with limited English ability and no keyboarding skills or previous experience with computers are eager to use them. This process may be seen as a significant aspect of technology transfer.

IMPLEMENTATING MICRO COMPUTER BASED CMI/MIS IN K-12 SCHOOLS

Michael Burger

Texas A & M University

The design and development of a computer managed instruction/ management information system (CMI/MIS) and the implementation of that system into the science, mathematics and social studies classrooms of three public schools is the focus of a project at Texas A&M University's Learning Technology Center. The proposed system is being designed (as a result of needs assessment activities conducted in each of the schools) to assist teachers in the development and evaluation of lesson plans and in the recording and evaluation of student performance (i.e., the automation of the record keeping and reporting functions associated with classroom instruction).

Pilot testing is scheduled for single user, single course environments during the Fall of 1985. A move to multi-user environments and multiple courses is scheduled to occur in 1986 and 1987.

Research efforts emphasize the development and maintenance of:

a) An appropriate user interface to link powerful, commercially available, multiuser, relational database software with teachers and administrators;

b) Educational databases containing information which is essential to effective instruction (i.e., objectives, resources, instructional activities, student, and teacher information);

c) Implementation strategies (including inservice training needs) required for successful integration and continued development of the system in various K-12 settings;

d) Software which could be used for research and development purposes to explore new relationships between data elements that are generated by the teachers as they use the system.

A status report of the project will be given as well as a description of scheduled activities. The project is being managed with the help of project management software, and participants will have opportunity to learn how well that approach to project management is working for this project.

COMPUTER-BASED LANGUAGE EXPERIENCE STRATEGIES FOR KINDERGARTEN AND FIRST GRADE STUDENTS

Claudia W. Cawthon, Barrow Elementary School

Thomas C. Reeves, The University of Georgia

The "Writing To Read" program, designed by John Henry Martin, supported by IBM, and evaluated by ETS, has received much national attention as an effective application of microcomputers in kindergarten and first grade classrooms. Bork (Personal Computers for Education, 1985) describes it as "one of the most ambitious curriculum development programs involving computers yet to be undertaken, perhaps the most elaborate". Early findings of the ETS evaluation involving over 15,000 students indicate positive results from the program in terms of improved reading, writing, and spelling skills. However, many school districts in this country will be unable to afford such an elaborate program or administrators may be unwilling to purchase programs which do not directly complement a district's investment in traditional "basal readers".

This presentation describes the efforts of one teacher in a small school system to teach creative writing/reading skills to five and six year-old children using Apple IIs, the Bank Street Writer word processing software, "invented spelling", and other "language experience-based" strategies. Analysis of the program over a one-year period indicates positive results in the development of reading and writing skills. The presentation will be supported by videotapes of classroom activities, examples of children's writing, and handouts describing instructional strategies.

THE FACULTY RETRAINING PROGRAM IN COMPUTER SCIENCE AT STEVENS INSTITUTE OF TECHNOLOGY

Lawrence E. Levine

Stevens Institute of Technology

Much has been written about the shortage of faculty who are qualified to teach computer science. One way of dealing with this shortage is to retrain faculty with expertise in other disciplines. Several programs have been developed with this goal in mind, and they have met with varying degrees of success. This talk will deal with the retraining program at Stevens Institute which began in June 1984.

This 15-month program leads to a Master of Science in Computer Science. The curriculum is highly structured given the requirement that participants who complete it are to have the wherewithal to teach all four years of undergraduate computer science. Participants attend special classes which are open only to those retraining. Exams, projects and homework are given on a regular basis. The N.J. Department of Higher Education is funding the program through a Loan Redemption Plan.

In addition to a discussion of the development of the program, its curriculum and thrust, this session will also outline the often surprising reactions of faculty when they again find themselves in a student role.

ON ENJOYING TEACHING ALGEBRA:

LETTING TEACHERS DO WHAT THEY DO BEST

John C. Miller

The City College of C.U.N.Y.

Elementary algebra instruction traditionally involves four steps:

1. Explain abstract principles.
2. Demonstrate via sample problems.
3. Assign similar problems.
4. Assess students' solutions.

Predictably, students do assigned problems by emulating sample problems, disregarding abstract principles, as emulation is more natural (and easier) than application of formal rules.

Algebra Mentor, an extensive program of computerized elementary algebra instruction being developed at CCNY, allows the instructor to concentrate on abstract principles by automating steps (2), (3) and (4). Algebra Mentor solves sample problems with simple step-by-step explanations, then poses a sequence of similar problems of increasing difficulty and monitors the students' solutions. It offers assistance at every step. It reviews prerequisites when necessary. It can be set to five different levels of difficulty. Thus, the instructor can postpone step (1), and discuss high-level abstractions with students who already understand concrete examples and low-level abstractions.

This method of instruction leaves the instructor more time for his essential functions: offering individual help to students with individual problems, and explaining abstract concepts ("What is an equation?" or "What is the distributive laws?") to students prepared to understand the answers.

Algebra Mentor will be demonstrated, its uses discussed, and its effectiveness analyzed.

TEACHER-"WHIZ-KID" PAIRING:

A COOPERATIVE MODEL FOR TECH ED TEACHER TRAINING

Lisa Novemsky

New Jersey Institute of Technology

In this experimental cooperative pilot program, New Jersey Institute of Technology, Digital Equipment Corporation, The Consortium in CAD/CAM Robotics and Video Vision Associates will cooperate with a local school system to provide in-service training in interactive video disc system, robotics, and teleconferencing for middle school teachers. These three computer-related new technologies, which can be accessed by the Apple IIE micro-computers already in the schools, will play a major role in facilitating the modernization of education. A unique aspect of this educational model is the pairing of each teacher with a top middle school computer "whiz-kid". In this recipe, the "whiz-kids" are the yeast....bubbling agents....raising the level of enthusiasm and innovative exploration for their new and seasoned teachers spreading the effects of the project with the rest of the school and larger community. This will maximize the achievement of both the teacher and student while integrating the human relations aspects of teaching into the program.

The program includes on-campus and at the middle school site activities from September through March. A teleconferencing network will keep the teams in constant communication with their university and industrial-based resources.

THE IMPACT OF COMPUTER-BASED COMPOSITION INSTRUCTION ACTIVITY ON THE FRESHMAN ENGLISH CURRICULUM: NEW TECHNOLOGY, NEW PROBLEMS AND NEW SOLUTIONS

Raj Singha

University of Cincinnati

Freshman English departments across the United States are beginning to integrate computer-based writing activity into the curriculum. The proposed presentation is designed to identify the nature of this transition and some of the problems this transition has brought about.

There are at least three components of this transition: 1) Student-writers in Freshman English programs are being trained to use microcomputers to conduct writing activity with the help of word processing; 2) an array of CAI programs, many of exceptional quality, have been developed by composition teachers to assist student-writers with their writing needs; 3) composition teachers are beginning to use advanced computer-based instructional techniques such as computer-based student-conferencing and computer-based collaborative writing activity.

In addition to a discussion of the computer-based instructional enhancements mentioned above, the presentation will also focus on the challenges of new technology for the composition teacher. In this regard, I will draw extensively from my experience as a composition teacher at the University of Cincinnati where I have been able to successfully integrate microcomputers into the Freshman English curriculum. This case-study approach will pinpoint the problems we have faced as well as share with our colleagues how we have solved them. The discussion will include issues such as training students in hardware and software use and yet continuing to focus on the writing process, and the issue of quality control in terms of the finished product - the student essay.

THE COMPUTER: TECHNOLOGICAL TEAMMATE FOR TEACHERS

Kay Walla

Miami University

American educators are in the process of providing leadership in converting from an industrial to an information-based society. Schools, colleges and universities have responded to the training challenge. Miami University (Ohio) has for five years been developing a computer education strand, beginning with making the faculty literate and moving to a computer science major and a computer education concentration. This session is designed to share some of the strategies used in the staff development process, as well as the techniques for incorporating computers into teacher education curriculum.

A search for appropriate uses of computers and for quality software is part of the challenge for teacher educators. This program will focus on ways to make effective use of computers in teaching and techniques for selecting and evaluating software. Reference and resource material handouts will be provided and the participants will have an opportunity to try some computer activities.

The focus will be on four major goals for teacher educators:

1. To be certain that all teacher education faculty are computer literate and computer users.

2. To provide computer applications for use in teaching every method's class regardless of subject area or discipline.

3. To assure opportunities for all teacher education candidates, as well as teachers in service, to be literate and successful computer users.

4. To provide a software critiquing technique for all teachers, both pre-service and inservice.

When these goals have been attained the computer will be truly recognized as a technological teammate for teachers. This session should aid in the process.

IMPROVING NURSES' TECHNICAL WRITING SKILLS
WITH COMPUTER ASSISTED INSTRUCTION

Robert Berner

University of Wisconsin at Oshkosh

In designing a computer-assisted instructional program in writing for nurses, both the basic problems of writing itself and the particular writing problems unique to nursing had to be considered. Usage, sentence structure and punctuation reveal differing degrees of difficulty in teaching basic skills by computer; nevertheless they can be taught. Furthermore, such a program is appropriate for teaching a basic organizational principle of technical writing: that readers most easily comprehend structures that are organized deductively rather than inductively.

The forms of writing chosen for such a program should be appropriate for student and professional nurses, but they should also illustrate how an audience can be analyzed to discover what organization and style are most appropriate for it. The forms chosen for the program under discussion were 1) the evaluation section of a care plan, 2) the resume and letter of application, 3) the description in an incident report, 4) instructions to patients, and 5) the journal article. Though the program must depend on examples of such forms, and though these are more appropriate as accompanying documents in the user's manual than on the screen, it is possible to explain the forms and to create considerable inter-play between user and computer in the process.

COMPUTER SKILLS FOR FUTURE ELEMENTARY TEACHERS

Lynda Thomas, Linda Thompson

Carroll College

The presentation would include the evolution and present status of the computer skills component of the elementary mathematics education program at Carroll College in Waukesha, Wisconsin. After studying other colleges' systems, surveying the current literature and conducting our own study of computer use in Wisconsin elementary schools, our college has developed a computer skills program which integrates computer use on a regular basis into our math content and math methods courses for elementary education majors.

The first phase of this program was started this year with the major emphasis on the evaluation of available elementary education mathematics software and its uses with the addition of introductory lessons in LOGO and BASIC. Next year we plan to continue this program and add weekly assignments in LOGO or BASIC which will tie into the topics of the week.

We tested this year's group at semesters beginning and end to determine gains in attitutde toward computers (using the Oetting Computer Anxiety Test), computer skills and knowledge (using our own test). We found gains in all areas with a significant gain in the area of attitude.

Objectives of Presentation:

1. To discuss some of the problems a small college faces when trying to develop such a computer skills program for elementary education majors.

2. To discuss the evolution and nature of Carroll College's program.

3. To present the results of our evaluation of our program during the 1984-85 school year.

PRO-NURSE A SOFTWARE PROGRAM FOR AUTOMATING INDIVIDUALIZED TESTING

Jean M. Arnold, William Greenhalgh

Rutgers The State University

Pro-Nurse is a computer software program, which was the outcome of team development by representatives of the Center for Computer and Informational Services and the College of Nursing at Rutgers University. The program includes a measurement instrument which analyzes an individual's diagnostic reasoning for a client situation.

The purpose of the program is to examine the diagnostic reasoning process by nursing students and/or nurses. Diagnostic reasoning includes assessment, objective formation hypothesis formation, intervention plan, criteria and rationale. Both information processing theory and decision theory serve as the theoretical framework for Pro-Nurse. The respondent identifies client problems, assigns priorities and designs a nursing care plan including criteria using a hand written format. Then these responses are entered into the computer and compared with the expert's performance. A printout of the individual's responses can be used to analyze their diagnostic reasoning process and to prescribe teaching interventions. These responses are subjected to group analysis using multiple regression.

SIMULTANEOUS COMPUTER LITERACY AND CAI AUTHORING

Lou Ann Emerson, Elizabeth Weiner

University of Cincinnati

The University of Cincinnati, College of Nursing and Health received a grant to implement a computer laboratory. Included in the grant was a proposal to develop CAI software as the number of quality CAI modules for nursing education was limited.

The development of CAI modules solely by the project directors was an unrealistic expectation. The project directors were faced with a commitment to CAI development and faculty who knew little about computers. The project directors, therefore, undertook development of faculty computer literacy and guidance of faculty authoring of CAI modules simultaneously. A variety of methods were used to achieve these goals. Initially, faculty development was directed toward introducing faculty to the potentials of CAI. Then faculty literacy programs were implemented. Faculty interested in working on CAI modules were recruited. Group meetings were held to discuss general areas of interest and to make broad plans for the projects. Teams were formed for specific projects.

The provision of guidelines, self-directed computer use packets, one-on-one interactions with the project directors, and special faculty workspace enhanced both literacy and CAI development. Within one year's time an interactive video project and several CAI modules involving eleven faculty were developed to the point of alpha testing.

ACADEMIC SOFTWARE - VISION AND PERSPECTIVE

Steven W. Gilbert

EDUCOM

The overall picture of current efforts to develop and apply computer software for academic purposes in higher education can best be described as a mess. Vendors view higher education as a chaotic fragmented market.

No single organization is capable of addressing the total confusing problem of academic software development and/or use. A conceptual framework is needed to permit important pieces of the problem to be identified and worked on separately by various organizations. Coordinating their efforts will be essential to avoid wasteful duplication. To help others address some of these problems more effectively, this presentation offers:

1. A brief sketch of a vision for the future impact of academic software in higher education.

2. A conceptual framework, including a set of software categories.

3. A system of distinct but interrelated mechanisms for enhancing and motivating the development, marketing, production, distribution, and use of academic software in higher education.

4. Some comments on the present and potential role of computer software in colleges and universities, with special emphasis on incentives, pricing structures, and the role of the faculty as developers and users.

A GENERIC TUTORIAL PROGRAM INCORPORATING VIDEODISC IMAGES AND TEXT PAGING

Ronald P. Jensh

F. Scott Beadenkopf

Jefferson Medical College

Some medical educators have avoided interactive video because of its apparent high cost and complexity. Therefore, an experimental program was written to explore the capabilities of an inexpensive IBM-PC/Pioneer LD-V1000 interface and to demonstrate the ability of the system to coordinate videodisc images with tutorial text and review questions.

As a first stage, a program was written to use the computer keyboard as a control pad for the videodisc player, enabling a non-programmer to play single frames or sequences of frames through single letter commands and frame numbers.

Secondly, a brief tutorial and set of review questions on blood cell classification was written and matched with appropriate slides from the "Hematology Slide Bank", a videodisc created by the Health Sciences Learning Resource Center at the University of Washington and distributed by Miles Pharmaceuticals. Using a word processing program on another computer, the text was written and formatted for effective presentation.

Third, a simple "authoring" program was written to count pages and lines and store the text and formatting data in a sequential file for program access.

Fourth, the presentation program was written. In it the tutorial text is read into memory and the user is prompted for the desired starting page number. All four pages of 80-column text video memory in the IBM-PC are used. The starting page is printed to video page one and displayed; the succeeding page is written to video page two and the preceding page to video page three. The appropriate videodisc image is found and displayed on a second monitor. A page flipping algorithm enables the user to page forward or backward through the text and supporting videodisc frames by pressing the "+" and "-" keys on the keyboard. The page flipping appears instantaneous. The user options menu is printed on video page zero and is always available.

The total system can be assembled for less than $4,000 and can be used to create an interactive shell for a videodisc slide collection.

A MICROCOMPUTER PROGRAM OF PULMONARY AND TISSUE GAS EXCHANGE

Joseph Boyle III

New Jersey Medical School

The student laboratory exercise has been eliminated or markedly reduced in many physiology curricula as well as in many other bioscience courses. This has occurred in response to many different pressures and has resulted in the student becoming largely a passive participant in the learning process in these courses. The present student is largely dependent on passively listening to lectures and studying static graphs and tables in textbooks. The student has also lost the laboratory experience which has traditionally been the format of learning the methods of deductive scientific reasoning

A number of microcomputer programs concerned with physiological simulations have been developed which encompass selected respiratory, cardiovascular and renal mechanisms. Some of these programs have been used for two years in place of scheduled laboratory exercises. These programs are used by small groups of students who follow an experimental protocol to obtain pertinent data. These activities occur with a faculty member present who can answer questions and help with any technical problems should they arise. The programs and the microcomputers are also available in the library for independent use by the students.

The acceptance of this type of teaching aid by the students and faculty will be discussed.

SIMULATIONS IN RESPIRATORY PHYSIOLOGY SUITABLE FOR USE IN MULTIPLE TEACHING SETTINGS

H. I. Modell

Virginia Mason Research Center and

University of Washington

The acronyms, CAI, CBE and CBT generally evoke a picture of a student working in an independent study mode. However, the portability and graphics capabilities of the microcomputer and the emergence of high quality video projectors have made it feasible to add the computer to the arsenal of visual aids for small group conference and large lecture hall settings.

We have revised our set of respiratory physiology simulations (Normal and Abnormal Lung Function - a program for independent study, Am Thorac Soc., 1975) to include pictorial outputs. The revised set is comprised of 12 programs covering pulmonary mechanics (4 models), gas exchange (6 models), chemoregulation (1 model) and acid-base balance (1 model). Each program displays data consistent with the user's variable values through one or more pictorials aimed at providing a schematic representation of the relevant physiology or a description of the model and how it is solved. Although an independent study "laboratory" manual is still included, the graphics enhancement makes the programs an extremely useful adjunct to small group discussions and "lectures" to large groups.

TEACHING A COMPUTER LITERACY COURSE IN A BACCALAUREATE NURSING PROGRAM

Patricia J. Morin

Nebraska Wesleyan University

This presentation describes the development and implementation of the course, Computers in Nursing and Health Care. The course is part of the curriculum in the baccalaureate nursing program at Nebraska Wesleyan University.

In the didactic portion of the course emphasis is placed on examining the applications of computers in nursing related to education, administration, research, and the health care system. In addition the historical development of the computer, an introduction to terminology, and principles of computers are introduced. "Hands-on" computer experiences are an integral part of the course. Results of the teaching strategies used in the "hands-on" experience were: a) increased level of comfort when using the computer, and b) increased proficiency of computer skills. The benefits derived by the students from the theory portion of the course included increased computer literacy and knowledge and skills needed to implement computers in the practice setting. The pros and cons of implementing the course are also included in the presentation.

THREE MODELS FOR DESIGNING PATIENT CARE SIMULATIONS FOR INTERACTIVE VIDEODISC

Jo Ann Reynolds

Human Resources Research Organization

Jo Ann Reynolds will discuss, compare and demonstrate three patient care simulations designed for presentation on a computer-controlled interactive videodisc system. The purpose of all the simulations is to provide continuing education by improving the decision making skills of critical care physicians and nurses.

In the first simulation the patient is the victim of a motor vehicle accident. The learner must assess the patient's injuries and provide appropriate treatment in "real time".

In the second case which involves respiratory management, the learner is responsible for weaning a cancer victim from a ventilator. The learner controls the time and adjusts the ventilator according to the patient's constantly changing condition.

The third patient care simulation is based on a diagnostic model. The learner must prioritize all possible diagnoses and systematically rule in or out each one. The progression of the case is based on the learner's selections or omissions.

EVALUATION OF A CAI MICROCOMPUTER LABORATORY

Elizabeth Weiner, Lou Ann Emerson

University of Cincinnati

The College of Nursing and Health received an Award for Program Excellence from the Ohio Board of Regents. The purpose of the award was to establish a 12 unit microcomputer laboratory which would serve to enhance the demonstrated excellence of the undergraduate nursing program. The challenge of the award was to document program enhancement as a result of the award.

The following criteria were used: documentation of faculty and student attitudes toward CAI, coordination of learning styles and microcomputer usage, anxiety scores of senior students toward state board examinations after CAI review, number of grants submitted in CAI, number of CAI modules authored, computer committee memberships, and faculty attendance at computer courses.

Post implementation, both faculty and junior students showed mean increases in their attitudes toward CAI, although not statistically significant. Faculty and Students' learning styles (Kolb, 1974) were found to be similar in nature and correlated highly with attitude toward CAI. Senior students had significantly decreased anxiety scores (Speilberger, 1970) in relation to the state RN licensure examination after using Moxby's NURSESTAR for review ($p < .001$). Other measures also demonstrated program enhancement after implementing CAI into the curriculum.

CONVERTING INSTRUCTIONAL VIDEOTAPES TO CAIV FORMAT

Dent M. Rhodes, Janet White Azbell

Illinois State University

This presentation deals with a process used to convert existing videotapes to a computer assisted interactive video instructional format. Effective CAIV is not simply video with a computer program built around it, as is often assumed. Because there are differences between CAIV instruction and instruction provided by conventional videotape, it is necessary as part of the conversion process to analyze the instructional format of the linear tape before a CAIV design can be formulated. CAIV design procedures can then be used to modify and adapt the instructional format to the needs of specific users.

The process of converting conventional videotapes to an interactive video format will be illustrated by the initial development phase of a cooperative effort between the Methodist Medical Center of Central Illinois (Peoria, Illinois) and faculty members of Illinois State University. The conversion process for videotapes of Continuing Medical Education presentations has been implemented in two steps: 1) formalizing a tape format analysis, which includes features to be observed in the tape production and addresses to roles which instructional designer and content specialist occupy, and 2) creating an instructional design specific to the learning approach familiar to physicians in continuing medical education applications.

Handouts of the specific model used to analyze videotapes in part of the conversion process and a demonstration of its application to medical tapes will be integral parts of this presentation.

PROGRAM PATH UTILIZATION OF NURSING STUDENTS USING A COMPUTER-ASSISTED TUTORIAL

Marianne E. Yoder

The purpose of this study was to identify how a group of baccalaureate nursing students utilized a computer-assisted instruction (CAI) tutorial. The majority of research in nursing CAI has been in measuring the outcomes of the CAI, and not in measuring how the learner progressed through the CAI program.

Data from 40 Junior baccalaureate nursing students were obtained using a CAI tutorial consisting of six sections. The subjects were presented a menu and allowed to select the sections and the order of presentation. Upon completion of each section, the subject was allowed to either select another section or to end the lesson. Each subject was also given a pre-test and post-test to measure learning achieved. Results revealed that 65% of the subjects completed all the sections of the lesson, but only 33% completed it in the recommended menu order sequence. A correlational t-test of the pre-test and post-test scores showed learning occurred, significant above the .001 level for all subjects, even those not completing all the lesson sections.

CONSISTENCY VS. MONOTONY IN INSTRUCTIONAL SOFTWARE

Marsha Young

National Technical Institute for the Deaf

at Rochester Institute of Technology

Consistency is a factor often forgotten on software review forms but a factor that impacts all users. When consistency is not a consideration in the development of the software, the result is that it is not there when it should be and is there when it should not be.

Consistency for completeness is an indicator of good development, can be accomplished using a content component checklist, and usually does not require identical ordering or presentation of content.

Speed of development and programming is often achieved by establishing a pattern in a prototype frame or lesson that is to be duplicated with varying content. Sometimes this pattern is established by the development team; sometimes by the lesson generating program being used. When the order, format, and wording of content become too predictable, the students' attention, motivation, and interest is lost. This results in content not read and more items answered incorrectly.

On the other hand, consistency can contribute to user friendliness. A user of a program should not have to guess nor work at remembering how to enter answers, what different cues or symbols mean, how to access different parts or options of the program, how to sign on or off, what to do next.

This discussion will examine the reasons for consistency and the places where it should be found.

GETTING INVOLVED WITH COMPUTERS
WITH LIMITED FUNDS AND WITHOUT A COMPUTER

Nadine Hackler

University of Florida

The time was right to get involved with computers. A proposal was written to secure funding for a computer, student programmer and technician. Partial funding was received, but did not include funding for hardware.

Several lessons were learned: be very specific about "their" involvement in the project when receiving outside funds, be realistic about time-line (takes twice as long as you think), have data base reviewed and completed before involving programmer to eliminate so many revisions, before beginning project know and understand institutions regulations and guidelines (was six months into the project before I saw these in printed form), and have a plan for how to complete if funds run out.

The purpose of the computer program on "Know Your Man-made Fibers" was to educate individuals, especially youth, about the characteristics, uses and care of man-made fibers. A pre and post-test was developed and used in the pilot testing to evaluate "how much was learned".

The objectives were for youth to understand the characteristics, uses, and care of man-made fibers as related to textile products; to be able to properly care for textile products; to be able to interpret and understand textile labeling.

TOTAL INTEGRATION

MOVING COMPUTERS INTO THE DEPARTMENT

Ann B. Kapp

Northeast Louisiana University

A report of how a department of home economics in a small state supported university joined the "computer age" incorporating computers into five major areas - administration, management, teaching, student use, and research - will be given.

Examples of computer utilization to be shown and described include, in the administrative area, student records and scheduling information; in the management area, budget and inventory control; in the teaching area, word processing, test construction, visual aid construction and classroom teaching aids; for the student use area, learning and evaluation materials. In the research area, examples of the ways computers have been incorporated in on-going projects and a proposal for an exciting state telecomputing project will be included.

The computer environment, the physical set-up including that for student, faculty and staff use will be described as well as methods for securing funds for software and hardware.

The third and final phase will be the training and motivational emphasis used for moving faculty from absolute zero use and knowledge to competence in the computer arena as well as a description of classes offered for majors in the various home economics curricula.

COMPUTER-BASED MODELS FOR DIETETICS EDUCATION

Lois Schroeder, Phyllis Kent

Syracuse University

This is a status report on the material presented in the May 1982 issue of JCBI describing interactive computer-based tutorial and problem-solving tools for nutrition and dietetics students enrolled in a course in diet therapy. The early version of this program was written in APL and was first implemented on the Sharp APL system at Syracuse University on an IBM 4341 computer. The program focuses on constructing diets for patients with renal disease which involves manipulating their protein, sodium, potassium, phosphate, and fluid intakes. After the student completes tutorial sessions for each of these areas, he is presented with computer-simulated, selective menu-writing problems for which he must choose food items to meet certain renal diet prescriptions. In the fall of 1984, the instructional units were revised and rewritten for use on microcomputers. The materials now take advantage of the microcomputer screen and graphic technology. The units will run on a stand-alone IBM PC or compatible microcomputer as well as a networked cluster of PCs. This presentation will show examples of recent instructional materials and describe the electronic classroom setting in which they are used.

COURSE PRODUCTION PROBLEMS ON MINI AND MICRO COMPUTERS

Jeffery Howell

Microtel Learning Services

This presentation shares with newcomers to the profession some of the author's experiences in producing course material on both mini and micro computers. Problems and solutions encountered while using McDonnell-Douglas' AIS-II, Hazeltine's Micro TICCIT, and the Wicat minicomputer authoring systems are cited. As are experiences using MS-DOS microcomputer authoring systems and languages.

Special emphasis is given to problems common to both mini and micro computer course production. Problems and solutions from the following topies are discussed:

a) transportability
b) documentation
c) template formation, macros, drivers, and "boiler plate"
d) evaluation, quality control and assurance
e) estimating and controlling costs and time
f) teams, roles and functions

LOOKING UP

A FUNDER'S PERSPECTIVE ON THE POST-SECONDARY USES OF TECHNOLOGY

Stephen C. Ehrmann

The Annenberg/CPB Project

There are a number of points where the national view is somewhat different from that of a single faculty member, or even a single college. I would like to describe some of the issues which strike me as most provocative.

1. Where can we find the vast sums of money needed for computer-based curriculum development (courses, exercises, utilities, tools, templates, etc.)? I would like to talk about the prospects for each of these factors.

2. Should foundations focus their efforts on developing "cutting edge" software that embodies the best new teaching ideas (in hope that those ideas will inspire others to design new software in other disciplines), or should we all focus our efforts on projects whose products will be most widely used? I will discuss the skimpy, anecdotal evidence.

3. What does a proposal look like from the funder's point of view? A key problem is "provincialism": the partly self-imposed isolation of the developer from other schools of thought. Some examples will be described, as well as some of the intellectual and political opportunities of investing in getting to know one another.

4. How do we attract more students to our college, using technology? Should we create a specialized unit for distance learning, using computers, television, telephone, tape and other tools to allow education from anywhere, at any time? Or, taking the opposite tack, should we seek to use information technologies on campus such that education becomes more accessible for everyone? I will compare strategies for pursuing each of these ends, and point out some of the important unanswered questions.

INTEGRATING PERSONAL COMPUTERS INTO A SUPERVISORY MANAGEMENT CURRICULUM

La Verne Ludden, Michael Miller

Indiana-Purdue University

Indiana-Purdue University at Fort Wayne offers an associate's degree and a bachelor's degree in supervisory management. This applied behavioral science program emphasizes the human relations aspects of management and seeks to prepare students for entry into supervisory or human resource development positions in manufacturing, government, health care, and other business and service organizations.

The department is embarking on a two-pronged approach to integrate personal computers into the curriculum. First, a group of new classes is being developed that will prepare students to use personal computers and application software in their careers. Second, an examination of the entire curriculum is being conducted to identify the appropriate use of personal computers in existing courses.

The presentation will describe the process of integrating personal computers into a supervisory management curriculum. The main points of the presentation will include the following:

1. The rationale for introducing personal computers into the curriculum of a supervisory management major will be examined.

2. The strategy for introducing personal computers to students will be reviewed. This includes a description of the approach which concentrates on the use of computer applications software rather than programming.

3. The rationale for infusing personal computers into the entire curriculum rather than just creating additional courses will be examined, and a description of this approach will be provided.

THE FUTURE OF CAI DEVELOPMENT BY UNIVERSITY EDUCATORS

Steve Hirst, Pete Trotter

CONDUIT

In 1980, Gerald Gleason reported on the use and development of CAI materials for microcomputers (Educational Technology, March, 1981, pp. 7-18). He found that the typical developer of CAI for higher education was a professor who got excited about CAI for use in his own class. However, producing the lessons was long and tedious, and frequently this effort was not rewarded professionally or financially.

Nearly five years later, the development of CAI for college and senior high schools is still difficult and unrewarding. Lack of skills and incentives causes many otherwise excellent projects to fail. Even if developers acquire substantial skills, their first CAI projects are frequently their last.

Because CONDUIT is concerned with the future of CAI development by university educators, we are engaged in several projects. To spread the costs of development we sponsor inter-university collaborations. To encourage the use of valid instructional strategies and user interfaces we supply programming tools and guides. To provide continuing support and encouragement we are investigating a faculty center to be jointly operated with the University of Iowa. Our presentation discusses the results of these projects, where more work is required, and our plans for the future.

INSTRUCTIONAL PRACTICE AND COMPUTER-BASED INSTRUCTION

AN IMPLEMENTATION MODEL

Marvin O. Heller, John F. Wedman

University of Northern Iowa

Before teachers can take full advantage of computer-based instruction (CBI), they must modify their instructional practices to accommodate the technology. Stevens (1980) suggested that two major factors affect implementing CBI: teachers' expertise with computers and teachers' attitudes toward computers. Most CBI-related teacher inservice programs focus on increasing teachers' computer expertise while paying little attention to teachers' attitude toward implementing the technology. Such an inservice emphasis is understandably given: 1) The large amount of information about CBI; and 2) the limited amount of information regarding teachers' attitudes toward CBI. Thus, a knowledge base focusing on teachers' attitudinal responses to CBI needs to be established.

This presentation reports on research studying teachers' concerns about CBI, describes how these concerns changed during inservice efforts, and integrates the research findings into a model for teacher inservice. The presentation addresses three specific objectives:

1. To describe the theory base and research methodology used to identify and monitor teachers' concerns about CBI.

2. To analyze concerns and implementation data collected during several computer-related inservice and research projects.

3. To develop a framework for designing and evaluating CBI-related teacher inservice.

Session participants will have the opportunity to critically examine their own concerns related to CBI. Presentation handouts will include a content summary, sample inservice instruments, and reference material essential for implementing the suggested inservice model.

MEETING FACULTY ACCESS TO INSTRUCTIONAL COMPUTING NEEDS

UTILIZATION TRENDS FOR THE UNIVERSITY OF FLORIDA'S FACULTY SUPPORT CENTER FOR COMPUTING

Wayne A. Wolfe

University of Florida

Two years of utilization data, accompanied by two faculty surveys, gauge how well the University of Florida's Faculty Support Center for Computing (FSC) is meeting faculty instructional computing needs and indicate future trends. As a product of a joint study between the IBM Corporation and the University of Florida, the FSC is beginning its third year of operation with the long range goal of providing all faculty easy access and appropriate training to use computing as an intellectual tool in their disciplines. One means for accomplishing this goal is instructional and training seminars in the use of computing, both personal computers and mainframe interactive services.

In May 1983, the FSC began offering a series of seminars for faculty. The curriculum originally resulted from faculty response to a survey and recommendations of the Final Report of the Task Force on the Use of Computers in Education. The personal computing seminars include introduction to the PC, word processing, introduction to databases, spreadsheets and BASIC. The mainframe interactive seminars include introduction to the Conversational Monitor System (CMS), Statistical Analysis System (SAS), and FORTRAN 77. Thus far, over 1,300 of the 2,400 fulltime, on-campus faculty who have potential access to the FSC have visited for at least one seminar.

A detailed analysis of utilization data and a Spring 1985 survey of FSC users and non-users clearly indicate future directions for an expanded curriculum. This expanded curriculum will meet the needs of a more sophisticated user group--those faculty who regularly visit the FSC--and attract many of the users who have yet to visit the FSC.

DEVELOPING EAR TRAINING SKILLS USING A MICROCOMPUTER

Richard Shirey

University of Akron

Two ear training programs present innovative microcomputer drill lessons for beginning music students.

Aural interval recognition has the student identify intervals presented melodically by the computer. Scores are based on a percentage for first-try answers. Scoring also allows the student as many tries as desired until he requests the answer. The instructor determines how many intervals are required to pass each unit. Eleven levels present all intervals from P1 through P8.

Melodic/rhythmic dictation has the student hear a melody and input the pitches (using paddle or keyboard) on a musicstaff. The computer supplies the key signature, time signature, and first note. All pitches must be correct before the rhythm, selected from note values on the screen, can be entered. Separate scores are kept for melodic and rhythmic dictation, and scoring is a percentage based on examples attempted versus those correct. Ten melodies must be accepted to record a score. Eleven levels present characteristic patterns including stepwise melodies, those incorporating all intervals, those beginning on tones other than the tonic, chord outlines, minor scale forms, chromaticism and modal melodies.

These programs make up a significant portion of the Aural/Oral Music Reading Skills course at the University of Akron.

A COMPUTER-AIDED DESIGN ENGINEERING CENTER

Donald H. Thomas

Drexel University

In June 1985 Control Data Corporation announced the CDC Engineering Center Network. The network's focus is improvement of engineering education and research, with emphasis on involvement of industry.

As part of this network, Drexel University has established a Computer-Aided Design Engineering Center, using a Cyber 180 with Macintosh workstations. The Center's focus is optimization of the design engineering process, from initial problem definition to the final manufactured output. The multi-disciplinary faculty and staff associated with the Center works closely with industry to define problems, gain necessary manufacturing expertise, and educate practicing engineers in new design techniques and other emerging engineering technologies. In addition, PLATO software is made available to industrial associates of the Center, as well as to the students at the University, through Macintosh workstations on campus and through modem connections to remote sites.

Drexel's principal commitments to CDC are 1) development of PLATO courseware for one or two lower-division engineering courses, 2) preparation of applications software packages for one or two upper-level courses, and 3) networking with other universities in the CDC Engineering Center Program to share engineering talent and ideas.

THE STUDENT INFORMATION RESOURCE SYSTEM

AN INTERACTIVE VIDEOTAPE PROJECT

Eric Halsey

University of Washington

The Student Information Resource System is to be installed at Seattle Central Community College in the fall of 1985. College students and other interested users will be able to use the system to get a variety of types of information about the college, ranging from the details of the admissions procedure to much broader topics such as "How Do I Choose a Career?"

The system runs on an IBM PC-XT serving IBM PC's via a 3-com Ethernet network. Each information station has a random-access VCR and a composite/RGB switchable monitor, as well as a PC. The system has been programmed using the Bell and Howell "MicroVisual Author (MVA) authoring system.

The 40-minute presentation will include a demonstration of the system, a discussion of the reasons for choosing videotape over videodisc for this project, a "report card" on the current release of the MVA authoring system, and a report on the performance and usefulness of the system as an information resource.

THE INTERACTIVE VIDEO INSTITUTE

A HANDS-ON APPROACH TO TEACHING INTERACTIVE VIDEO

David Burke

Hocking Technical College

Interactive video is an effective but complex teaching method requiring a producer to have knowledge of instructional design, video production, and computer programming. The growth of interactive video is clearly dependent on the availability of a large pool of quality software. In order for more software to be available, many more individuals must be trained in the actual production of interactive video programs. Currently there are a number of training seminars and workshops available which claim to provide training in interactive video design and production, yet few actually use a hands-on approach which gives the participants the experience to fully understand the potential and pitfalls of producing in the medium.

Hocking Technical College has been producing and using interactive video for three years, and we were frequently asked to help others get started, so that they could shorten their learning curve. As a result, we have developed a 4 1/2 day, hands-on, Interactive Video Institute, originally designed for nurses, in which the participants design a program, use an interactive video authoring system, and actually experience the editing of their video. In the end, participants show their productions to each other in a "Grand Premier". For practical reasons, the Institute uses videotape as the interactive medium, but portions of the seminar focus on working with videodiscs as well.

This presentation will describe the background and rationale for taking this teaching approach, the actual seminar itself, and the successes and failures of this method. Actual teaching materials used in the Institute will be available, and samples of the participant's final efforts will be shown. Recommendations will be presented for improving the delivery of training in interactive video production.

INTEGRATING ORIENTING STRATEGIES
IN THE DESIGN OF INTERACTIVE VIDEO

Michael J. Hannafin, Pennsylvania State University

Curtis Hughes, Grumman Aerospace

The purposes of this presentation will be to extrapolate relevant research on the effects of orienting strategy specificity to the design of interactive video, and to describe and illustrate a case study where orienting strategies were systematically applied to obtain desired effects.

The case study where various orienting strategies were applied was an interactive video lesson designed to teach a variety of concepts related to rocketry and propulsion. The lesson was designed for use by college level engineering students, but the concepts could also be acquired readily by motivated high school students. Several types of learning were required, including verbal information, procedural and problem-solving. Orienting strategies were differentiated according to the type of required learning. Some were presented verbally either using computer text or audio, some were presented as graphic organizers, while others were presented via video. In each case, the empirical basis for the orienting strategy given the nature of the learning task were identified.

A MULTI-LEVEL GUIDED PRACTICE TUTORIAL

USING INTERACTIVE VIDEO

John F. Moore

Virginia Polytechnic Institute and State University

This presentation will describe how an interactive video program was designed to provide both tutorial instruction, guided practice, and job aids. The talk will emphasize content organization, user interaction, screen design, and graphic support.

The pilot program teaches undergraduates the basic operation of a Compugraphic computerized typesetter. Tutorial segments use split-screen video with computer text and graphics. After each tutorial, the interactive system guides the students in practicing various procedures. The program also provides a variety of help screens and video segments for clarification. The material is designed for different uses by both beginning and advanced students.

The instructional design process involved a series of task analyses and walkthroughs. Early versions of the tutorials and practice exercises were verified by observing and questioning a novice operator. Interaction features were designed to emphasize ease of use in both planned settings. Several learner control features were included to improve access to the information and job aids. Graphics and captions were used to emphasize particular typesetting CRT system messages and keyboard sequences.

FOCUS ON L.O.C.I.S.

INTERACTIVE VIDEODISC TRAINING FOR LIBRARY USERS

Thomas C. Reeves, William D. Aggen

MetaMedia Systems, Inc.

This interactive videodisc training program was designed to instruct researchers and library patrons in the use of the Library of Congress Information System (LOCIS), an automated computer catalog search system. Ever since the Library of Congress began to switch from a card-based catalog to a computer-based catalog in the 1960's, there have been increasing complaints that the computerized search procedures are difficult to learn and use. Some editorials and letters to the editor in the Washington Post and other periodicals have claimed that the new systems hinder rather than assist the work of scholars and other users of the world's largest library. "Focus on L.O.C.I.S." is one part of the Library of Congress' efforts to make the computerized catalog a "user friendly" experience. The program uses analogies, tutorials and simulations to bridge the gap between traditional card catalog searches and the automated computer search. Patrons are encouraged to use the interactive videodisc program on their own, but professional librarians are always available to help. This paper describes the project and the instructional design strategies of "Focus on L.O.C.I.S." The presentation will be supported through demonstrations from the videodisc.

AN OVERVIEW AND PLAN FOR THE CANADIAN IMPLEMENTATION OF COMPUTER ASSISTED LEARNING COURSEWARE FOR SPECIAL NEEDS CHILDREN, ADOLESCENTS AND ADULTS

Lauran Sandals, University of Manitoba

This study was undertaken on behalf of the Department of Communications, Government of Canada, Ottawa under the University Research Grants programme. This report presents 1) the past research in the area of computer assisted learning for special needs individuals worldwide, 2) the present research and progress developing computer assisted learning materials for special needs individuals, and 3) a proposal for future research and curriculum development in the area of computer assisted learning for special needs individuals from the Canadian perspective; and a nationwide implementation policy for future federal and provincial funding in this area. A three-factor model for the implementation of this report was developed consisting of degree of need - physically, developmentally, handicapped, etc.; learning from, with and about computers and courseware/lessonware that can run on stand alone commercial hardware or packages that need new hardware or interface developments. The organization of the national policy focuses in on a) centers of excellence and research, b) a national dissemination network, and c) ongoing ad hoc committees/think tanks.

A NEUROCOGNITIVE APPROACH TO TEACHING READING

A COMPUTER ASSISTED PROGRAM

Phyllis Anne Teeter, Philip Smith, Randall Ryder

University of Wisconsin

The purpose of this presentation is to discuss and present a set of software reading programs for children with reading disabilities. The microcomputer programs were developed using the most recent neuropsychological and cognitive processing theories to investigate if reading skills improve when instruction matches the child's cognitive processing strengths.

Three microcomputer reading programs were developed for third and fourth grade reading levels, reflecting different processing strategies, including: 1) a program that combines simultaneous and sequential processing (typical of traditional approaches which do not emphasize one strategy; 2) a program that emphasizes simultaneous strategies; and 3) a program that emphasizes sequential processing. The reading programs consist of 10 weeks of instruction, with four half hour lessons per week. The lessons include four major tasks to increase vocabulary knowledge and reading comprehension.

CAI GUIDELINES FOR DIALOGUE INTERACTION

Aaron B. Navarro

The MITRE Corporation

With the proliferation of microcomputers and CAI in our educational institutions, it has become imperative to set formal guidelines for developing effective courseware. Specifically, there is a need for using scientific methods and approaches to develop CAI guidelines for dialogue interaction and apply them in an adaptive way according to the traits and aptitudes of the learner.

The research has utilized experimental and clinical methods for gaining additional insight into the individualized nature of CAI and for investigating the nature of the relationship between CAI dialogues (system-guided and student guided) and the learner's cognitive style and locus of control. Three principal research hypotheses were tested:

1. RH1: By first introducing externally-oriented or field dependent students to the system-guided mode and then to the student-guided mode, they will prefer the student-guided mode; whereas if they are first introduced to the student-guided mode, they will prefer the system-guided mode.

2. RH2: Students who are both internally oriented and field independent will prefer the student-guided mode regardless of sequence in mode exposure.

3. RH3: Students will perform better in their preferred mode of interaction by scoring higher and completing a problem-solving task quicker.

CAI GUIDELINES FOR DIALOGUE INTERACTION

Aaron B. Navarro

The MITRE Corporation

With the proliferation of microcomputers and CAI in our educational institutions, it has become imperative to set formal guidelines for developing effective courseware. Specifically, there is a need for using scientific methods and approaches to develop CAI guidelines for dialogue interaction and apply them in an adaptive way according to the traits and aptitudes of the learner.

The research has utilized experimental and clinical methods for gaining additional insight into the individualized nature of CAI and for investigating the nature of the relationship between CAI dialogues (system-guided and student guided) and the learner's cognitive style and locus of control. Three principal research hypotheses were tested:

1. RH1: By first introducing externally-oriented or field dependent students to the system-guided mode and then to the student-guided mode, they will prefer the student-guided mode; whereas if they are first introduced to the student-guided mode, they will prefer the system-guided mode.

2. RH2: Students who are both internally oriented and field independent will prefer the student-guided mode regardless of sequence in mode exposure.

3. RH3: Students will perform better in their preferred mode of interaction by scoring higher and completing a problem-solving task quicker.

SCREEN DESIGN TECHNIQUES USING THE PILOT PROGRAMMING LANGUAGE

Karen A. Droms

NCR Corporation

This presentation will explore good screen design techniques for CAI programs, and how the PILOT language can be used to implement those techniques. NCR PILOT II will be used for all demonstrations and discussions. While this is the version of PILOT marketed by NCR, it is based on the industry standard common PILOT, differences between NCR PILOT II and others (Super PILOT, IBM PC) will be documented in the presentation.

One accepted method of information presentation in CAI is through the use of information "windows", areas of the screen in which the same information is presented continuously throughout the lesson, and that remain approximately the same size throughout the lesson. These include directions windows, orientation (or title) windows, text windows, error message windows, and graphics windows. The order in which information is presented in these windows is also important, since a learner's perception affects how he or she learns.

Through the use of PILOT's viewport commands and execute indirect commands, and some standard coding techniques, the use of information "windows" is relatively simple. By employing a standard algorithm for information presentation, designing CAI screens and implementing them in a PILOT program becomes easy to do.

TEACHING PILOT IN CAI ACADEMIC COURSES AND WORKSHOPS

W. Kirk Richardson

Georgia State University

The increasing need for CAI requires that more people be trained to work on CAI production teams. This should include both the education of students in the universities and retraining of professional trainers and educators.

Frequently the people with the best instructional design skills and understanding of the educational process have little or no computer programming skills and may be reluctant to use a computer. This paper addresses the problem of teaching these people some basic skills of CAI production in a short period of time.

The PILOT language is well suited for this purpose as basic PILOT programming can be learned relatively quickly and PILOT can be taught using instructional examples which these people understand.

The PILOT course described in this paper meets the following requirements:

1. The instruction is hands-on.
2. No computer knowledge is assumed.
3. The examples have instructional relevance.
4. The examples show aspects of good instructional design.
5. There is frequent feedback and success.
6. Students may proceed at their own pace in the class.
7. Little "computerese" is used.
8. After the course the student may continue to use and learn about PILOT with only a PILOT reference manual for support.

THE EFFECTS OF ORIENTING STRATEGIES AND PRACTICE ON THE LEARNING OF CUED AND UNCUED INFORMATION

Michael Hannafin

The Pennsylvania State University

Orienting strategies are techniques introduced prior to lesson segments that aid in the integration of forthcoming instruction. Different kinds of orienting strategies influence learning in very different ways. In many cases, specific orienting strategies such as behavioral objectives and embedded pre-questions improve the learning of intended, but inhibit the learning of incidental, information (Hamilton, 1985). The learning of both intended and incidental, as well as increasingly abstract learning, may also be improved through the utilization of abstract orienting strategies.

Two other variables which have been examined extensively in non-computer studies are cuing and practice. Cuing refers to directing learner attention differentially to given aspects of a lesson.Like orienting strategies, cues can be explicit or general. Appropriate use of cuing, especially the relationship between instructional cuing and cue-relevant practice, is important in the design of computer-based instruction. The effects of cuing versus practice, and the explicitness of the cues, on factual recall was the focus of this study.

AUTHORING SOFTWARE AND EFFECTIVE COMPUTER ASSISTED INSTRUCTION

Cindy Leshin, Joellyn C. Pollock

Arizona State University

Computers have received a great deal of attention as a new technology with the capability to improve the effectiveness of instruction. However, much of the earlier instruction connected with computers merely imitated other instructional media or failed to incorporate instruction design principles. Research has shown that all learning benefits attributed to CBI are readily available from other media.

Authoring software developed in the past few years makes it possible for the instructional designer to create effective computer assisted instructional programs using instructional design principles. The problem with these authoring programs, however, is that the systems vary in the ease with which the developer can implement these instructional design principles.

In this presentation authoring software (TenCore, McGraw Hill, Super PILOT and QUEST) will be reviewed. Instructional design strategies for computer assisted instruction will be discussed in relationship to the strengths and weaknesses of each of these authoring systems.

THE ACCURACY OF STUDENTS' COGNITIVE MONITORING DURING COMPUTER-BASED INSTRUCTION

Casey Garhart

The Pennsylvania State University

Individuals process information in different ways. The strategies preferred by one learner will differ from those preferred by another. One way to accomplish individual tailoring of instruction is by transferring control of the lesson's structure and sequence to the learner. The greater the extent to which the locus of instructional control rests with the learner, the more individualized the instruction should become.

Unfortunately, research in this area has failed to demonstrate that learner control improves learning. Instead, in most studies subjects who had control of their instruction terminated the instruction too soon.

One possible explanation for this finding is that many learners may not assess accurately their understanding of lesson information. If instructional control decisions are made based upon accurate student comprehension monitoring we should expect successful outcomes; if based upon student comprehension monitoring that is ineffective, however, the outcomes of learner-directed lessons are likely to be poor. The accuracy of student comprehension monitoring, therefore, is a critical variable in learner directed computer-based instruction.

This presentation will report on a study designed to investigate how students monitor their own understanding. The study examined the correlation between students' self-assessed understanding of factual and inferential material during a CAI lesson, their actual performance on embedded questions during the lesson and on a posttest, and the time required to answer each question.

ENHANCING STUDENTS CREATIVE ABILITIES THROUGH THE USE OF THE MICROCOMPUTER

J. Mark Tisone

Old Dominion University

A study recently completed at The Ohio State University has examined the question of whether middle school students creative abilities of fluency, flexibility, originality and elaboration can be enhanced through the use of microcomputer technologies (e.g., simulations, videodisc, application packages) as well as traditional program (Productive Thinking Program, Purdue Creative Thinking Program, Parnes Program) approaches.

From a review of research concerning traditional creative development programs, numerous methodological and conceptual problems have been identified. Therefore, it was hypothesized that those students who participated in the microcomputer based program would score significantly higher on the fluency, flexibility, originality, and elaboration levels of the Torrance Tests of Creative Thinking (a creative achievement test) given at the conclusion of the study than those who participated in the traditional methods of acquiring creative skills and behaviors.

This study provided an opportunity to compare two distinct programs under the same conditions and settings. Most studies of creativity have reported significant results, but were primarily based on a pretest posttest group design where no control was ever used.

The study incorporated the fact that microcomputers have proved to be a useful and valuable tool for today's educators. However, no study to date has tried to evaluate the effect of microcomputers on students' creative abilities. This study has gathered evidence that may ultimately change the way creative development exercises and activities are presented to both primary and secondary students.

THE FIPSE TECHNOLOGY STUDY GROUP
AND THE EDUCATION TECHNOLOGIES DATABASE

Mary Grace Smith

Harvard University

The Educational Technologies Database Project supported by a grant from the Fund for the Improvement of Post-Secondary Education (FIPSE) is compiling a directory of information on the uses of microcomputers in educational settings nationwide. The database will be published electronically on Compu-Serve, a Consumer Information Service, and will be searchable.

Data will be provided about instructional, administrative and research applications from pre-school through post-secondary institutions, both public and private, and in non-institutional settings as well. In addition to statistical information, the database will also be a compendium of the latest advances in educational technology in real life settings and will provide ready access to knowledgeable colleagues around the country, thus encouraging collegial consultation and networking among educators.

As a result of our research and as a result of participating in the Fipse Technology Study Group (FTSG) we have gathered information on a variety of exceedingly interesting computer based instructional projects in higher education. FTSG is engaged in an experimental computer conference project on the New Jersey Institute of Technology EIES network also aimed at increasing contact among computer using educators.

AN HIERARCHICAL, MENU-DRIVEN MICROCOMPUTER NETWORK

G. Brink, L. Glasser

University of the Witwatersrand

A microcomputer network for CAI, based on a ProNET local-area network with 10Mb Corvus hard disk, has been implemented using Apple microcomputers as terminals; about 25,000 programs are run on the network each year.

The software resource consists of six program sub-systems, each containing up to 80 programs. When the system is accessed, a menu is loaded, which offers the various sub-systems. Within each sub-system, lower-level menus offer individual volumes from the hard disk, and then programs within each volume. The individual menu programs are all identical except for the options offered; a master program permits generation of menu options at will.

When a user exits a program, a grade is recorded against his name on a unique volume of the disk. The user is then returned to his original volume menu, from which he may select other programs on that volume or, optionally, return successively to higher-level menus, until he reaches the initial menu of sub-systems.

Needed utility routines are loaded by the main sub-system menu into the lower memory of the Apple. The user's record is also stored in lower memory, for reference by programs when needed. Programs themselves are loaded into upper memory only.

PRO-NURSE: A SOFTWARE PROGRAM FOR AUTOMATING INDIVIDUALIZED TESTING

Jean M. Arnold, William Greenhalgh

Rutgers State University

Pro-nurse is a computer software program, which was the outcome of team development by representatives of the Center for Computer and Informational Services and the College of Nursing at Rutgers University. The program includes a measurement instrument which analyzes an individual's diagnostic reasoning for a client situation. The purpose of the program is to examine the diagnostic reasoning process by nursing students and/or nurses. Diagnostic reasoning includes assessment, objective formation hypothesis formation,intervention plan, criteria and rationale. Both information processing theory and decision theory serve as the theoretical framework for Pro-nurse. The respondent identifies client problems, assigns priorities and designs a nursing care plan including criteria using a hand written format. Then these responses are entered into the computer and compared with the expert's performance. A printout of the individual's responses can be used to analyze their diagnostic reasoning process and to prescribe teaching interventions. These responses are subjected to group analysis using multiple regression.

EVALUATION OF A CAI MICROCOMPUTER LABORATORY

Elizabeth Weiner, Lou Ann Emerson

University of Cincinnati

The College of Nursing and Health received an Award for Program Excellence from the Ohio Board of Regents. The purpose of the award was to establish a 12 unit microcomputer laboratory which would serve to enhance the demonstrated excellence of the undergraduate nursing program. The challenge of the award was to document program enhancement as a result of the award.

The following criteria were used: documentation of faculty and student attitudes toward CAI, coordination of learning styles and microcomputer usage, anxiety scores of senior students toward state board examinations after CAI review, number of grants submitted in CAI, number of CAI modules authored, computer committee memberships, and faculty attendance at computer courses.

Post implementation, both faculty and junior students showed mean increases in their attitudes toward CAI, although not statistically significant. Faculty and students' learning styles (Kolb, 1974) were found to be similar in nature and correlated highly with attitudes toward CAI. Senior students had significantly decreased anxiety scores (Speilberger, 1970) in relation to the state RN licensure examination after using Moxby's NURSESTAR for review ($p < .001$). Other measures also demonstrated program enhancement after implementing CAI into the curriculum.